

Titre: Méthodes itératives pour la résolution, par éléments finis, du
Title: problème de stokes non linéaire

Auteur: Jean-Philippe Marcotte
Author:

Date: 2000

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Marcotte, J.-P. (2000). Méthodes itératives pour la résolution, par éléments finis,
Citation: du problème de stokes non linéaire [Master's thesis, École Polytechnique de
Montréal]. PolyPublie. <https://publications.polymtl.ca/8852/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8852/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

MÉTHODES ITÉRATIVES POUR LA RÉOLUTION, PAR
ÉLÉMENTS FINIS, DU PROBLÈME DE STOKES NON
LINÉAIRE

JEAN-PHILIPPE MARCOTTE
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
JUN 2000



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-57419-9

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

MÉTHODES ITÉRATIVES POUR LA
RÉSOLUTION, PAR ÉLÉMENTS FINIS, DU
PROBLÈME DE STOKES NON LINÉAIRE

présenté par: MARCOTTE Jean-Philippe

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. LAFLEUR Pierre , Ph.D., président

M. FORTIN André , Ph.D., membre et directeur de recherche

Mme HEUZEY Marie-Claude , Ph.D., membre

À mes parents, Cécile et François.

Remerciements

Je tiens à remercier tout particulièrement mon directeur de recherche, André Fortin pour son intérêt et son soutien tant technique qu'intellectuel et qui m'a permis d'accomplir cette maîtrise. Également, je remercie Robert Guenette, pour son aide très précieuse.

Je remercie M. Pierre Lafleur et Mme Marie-Claude Heuzey d'avoir accepté de faire partie de mon jury.

J'en profite pour remercier mes collègues qui ont su agréementer mon séjour lors de mon passage à l'École Polytechnique: Steven et Antoine pour leur aide avec *LaTeX*, Bass, Donatien, Mitra, Gérardo, Jean-Luc, Alain Lioret et Alain Béliveau.

Je veux aussi souligner l'aide des membres du *GIREF*. Plus particulièrement, Carl Robitaille pour ses nombreux conseils à propos de **MEF++**.

Finalement, je voudrais exprimer toute ma gratitude envers mes parents pour leur amour et leur soutien qui m'ont été si chers.

Résumé

Ce mémoire est consacré à l'étude d'écoulements stationnaires de fluides newtoniens généralisés incompressibles. Les fluides newtoniens généralisés, contrairement aux fluides newtoniens, ont une viscosité qui dépend du taux de cisaillement auquel ils sont soumis. On présente donc deux modèles pour représenter cette dépendance, soit la *loi d'Ostwald-De Waele*, aussi connue sous le nom de *loi de puissance* et la *loi de Carreau*. Nous nous proposons donc de résoudre les équations de Stokes pour les fluides incompressibles avec la viscosité modélisée à l'aide des modèles précédents. Étant donnée cette dépendance entre la viscosité et le taux de cisaillement, les équations de Stokes sont, dans le cas de fluides newtoniens généralisés, non linéaires. Nous utilisons la méthode de Newton pour contourner ce problème de non-linéarité.

La discrétisation des équations de Stokes est faite à l'aide de la méthode des éléments finis. On présente d'ailleurs une revue assez exhaustive de la méthode des éléments finis spécifiquement appliquée à la résolution du problème de Stokes. On essaie de présenter la méthode d'un point de vue assez informatique pour qu'il soit facile de faire le lien entre le développement théorique et ce qui est programmé dans un ordinateur. La résolution des systèmes d'équations provenant de la discrétisation par éléments finis est faite à l'aide de la méthode d'Uzawa qui permet de résoudre les problèmes en vitesse et en pression séparément. La résolution de ces deux sous-problèmes est effectuée par la méthode itérative du gradient conjugué. Toujours dans le but de lier la théorie et l'informatique, on discute de la notion de *Matrix-Free* qui est utilisée en pratique pour résoudre des systèmes d'équations sans connaître explicitement la matrice du système. On pourra ainsi résoudre un système du type $BA^{-1}B^T X = G$ sans avoir à calculer l'inverse de la matrice A .

L'accent, dans ce mémoire, a donc été mis sur la résolution de problèmes tridi-

mensionnels. D'ailleurs tous les problèmes que nous avons résolus, l'ont été sur des géométries en trois dimensions. On présente deux problèmes tests d'écoulement sur des géométries très simples, dont un écoulement de *Poiseuille* dans un tube, ce qui permet de comparer notre solution à la solution analytique qui est connue et ainsi valider notre code. On donne également quelques solutions sur des géométries plus complexes. Finalement, on tente d'appliquer notre méthode à un problème d'écoulement directement tiré de l'industrie des polymères. On introduit brièvement un problème d'extrusion pour ensuite décrire l'écoulement d'un fluide dans des vis d'extrusion. Bien que le problème présenté dans le mémoire n'ait pas réellement de signification physique, il permet tout de même de voir le comportement de nos méthodes face à des maillages plus compliqués.

En annexe, on retrouve quelques rappels sur les matrices définies positives puisque la méthode du gradient conjugué exige d'avoir de telles matrices. Ensuite, on trouve un rappel de la méthode du gradient conjugué pour résoudre des systèmes d'équations. On retrouve également une description complète du code d'éléments finis qui nous a permis de résoudre notre problème de Stokes, code que nous avons élaboré avec l'aide des membres du *Groupe interdisciplinaire de recherche en éléments finis*.

Abstract

This master's thesis is dedicated to the study of incompressible generalized Newtonian fluids. Generalized Newtonian fluids have a viscosity that depends on the strain rate. We present two models to express this relation, the *Ostwald-De Waele's law*, also known as the *power law* and *Carreau's model*. We tried to solve Stokes' equations for incompressible fluids with viscosity modeled with the previous two models. Due to the dependence between viscosity and strain rate, Stokes' equations are, for generalized Newtonian fluids, non linear. So we use Newton's method to resolve this problem.

Discretization of Stokes' equations will be done with the finite element method. We give a complete review of this method specially applied to solve Stokes' equations. We use the Uzawa's method, combined with the conjugate gradient method, to solve the equations systems provided by finite element method.

We focus on tridimensionnal applications, which are now tractable due to the improvement of computers in the recent years. First, we present a *Poiseuille's flow* in a tube and compare the numerical solution to the analytic solution which is known in order to validate our code. Next, we solved Stokes' equations on a cavity. We also give some solutions on more complex geometries, such as the contraction . Finally, we applied our method to the flow in a extruder screw. Even if this flow does not have a quite real signification, it helps to see the comportment of our methods on more complex meshes.

Table des matières

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	ix
LISTE DES FIGURES	xi
LISTE DES TABLEAUX	xiii
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
LISTE DES NOTATIONS	xv
LISTE DES ANNEXES	xviii
INTRODUCTION	1
CHAPITRE 1 ÉQUATIONS DE LA PHYSIQUE	4
1.1 Équations de la mécanique des fluides	4

1.2	Modèles pour la viscosité	7
CHAPITRE 2 MÉTHODE DES ÉLÉMENTS FINIS		11
2.1	Formulation variationnelle des équations de Stokes	12
2.2	Linéarisation (méthode de Newton)	14
2.3	Discrétisation du problème de Stokes	17
2.4	Assemblage de la matrice globale	24
CHAPITRE 3 RÉOLUTION DU SYSTÈME GLOBAL		26
3.1	Méthode d'Uzawa	26
3.2	Autres approches de résolution	32
3.2.1	Pénalisation	32
3.2.2	Stabilisation	32
3.3	Visualisation des résultats	33
CHAPITRE 4 RÉSULTATS NUMÉRIQUES		34
4.1	Validation des résultats	34
4.1.1	Écoulement de <i>Poiseuille</i> dans un tube	34
4.1.2	Écoulement dans une cavité cubique	38
4.2	Autre géométrie	42
4.3	Problèmes rencontrés en industrie	44
CONCLUSION		50
RÉFÉRENCES		52
ANNEXES		54

Liste des figures

1.1	Exemple d'un domaine Ω en 3D	4
1.2	Viscosité en fonction du cisaillement pour différents fluides	8
2.1	Exemple d'un maillage du domaine Ω en 3D	11
2.2	Passage d'un élément K à l'élément de référence \hat{K} en 2D	21
2.3	Passage d'un élément K à l'élément de référence \hat{K} en 3D	21
2.4	Élément triangulaire P2-P1	22
2.5	Élément tétraédrique P2-P1	22
2.6	Élément triangulaire <i>Mini</i>	23
2.7	Élément tétraédrique <i>Mini</i>	23
4.1	Maillage d'un tube avec 6000 éléments (tubex2)	35
4.2	Solution numérique d'un écoulement de <i>Poiseuille</i> sur le tube	36
4.3	Solution dans le plan $z = 0$ de l'écoulement de <i>Poiseuille</i> sur le tube .	36
4.4	Solutions numériques et solution analytique d'un écoulement de <i>Poiseuille</i>	37
4.5	Maillage d'une cavité avec 384 éléments	38
4.6	Solution numérique d'un écoulement induit dans la cavité (solution obtenue avec 3072 éléments)	39
4.7	Solution numérique d'un écoulement induit dans la cavité (solution obtenue avec 24576 éléments)	40

4.8	Isovaleurs de la composante x de la vitesse, dans la cavité	40
4.9	Norme infinie du résidu en fonction des itérations dans le cas linéaire	41
4.10	Maillage d'une contraction avec 30720 éléments	42
4.11	Solution numérique d'un écoulement dans la contraction	43
4.12	Isovaleurs de la pression, dans la contraction	43
4.13	Schéma d'une extrudeuse avec une monovis	44
4.14	Maillage entre les dents d'une vis à extrusion - Vue de face	45
4.15	Maillage entre les dents d'une vis à extrusion - Vue de biais	46
4.16	Solution de l'écoulement dans la vis, dans différents plans	47
4.17	Solution de l'écoulement dans la vis, dans différents plans (suite)	48
4.18	Solution de l'écoulement dans la vis	49

Liste des tableaux

4.1	Informations à propos des maillages tube, tubex2 et tubex4	37
4.2	Informations à propos des maillages cube, cubex2 et cubex4	39
4.3	Informations à propos du maillage Contraction	42
4.4	Informations à propos du maillage Vis	46

Liste des sigles et abréviations

CERCA	Centre de recherche en calcul appliqué
GIREF	Groupe interdisciplinaire de recherche en éléments finis
GMRES	Generalized minimal residual
MINRES	Minimal residual

Liste des notations

Caractères usuels

\tilde{A}	matrice des contributions linéaires en vitesse
\hat{A}	matrice des contributions en vitesse provenant de la linéarisation
A_0	matrice de toutes les contributions en vitesse
\mathcal{A}_k	matrice pour l'algorithme d'Uzawa
B	matrice des contributions de l'opérateur de divergence
B^T	matrice des contributions de l'opérateur gradient
C	constante pour l'écriture des modèles de viscosité
d	dimension dans laquelle se situe le problème
\mathbf{f}	forces prescrites sur le domaine physique
F	matrice des contributions relatives aux forces \mathbf{f}
\mathbf{g}	fonction relative aux conditions aux limites de <i>Dirichlet</i>
G_k	vecteur pour l'algorithme d'Uzawa
\mathbf{h}	fonction relative aux conditions aux limites de <i>Neumann</i>
I	tenseur identité
J	matrice de préconditionnement de Jacobi
K	un élément
\hat{K}	l'élément de référence
L	longueur
M	matrice masse
\mathbf{n}	vecteur normal extérieur
n_D^u	nombre de degrés de liberté en vitesse
n_D^p	nombre de degrés de liberté en pression
n_c^u	nombre de noeuds de calcul en vitesse sur l'élément K
n_c^p	nombre de noeuds de calcul en pression sur l'élément K

N	indice de pseudoplasticité
p	pression
P_k	vecteur pression à l'itération k
q	fonction test pour l'équation de conservation de la masse
r	position en coordonnées polaires
R	rayon
\mathbf{R}	l'ensemble des réels
\mathcal{R}_1	résidu pour l'équation de conservation de la quantité de mouvement
\mathcal{R}_2	résidu pour l'équation de conservation de la masse
R_1	résidu vectoriel pour l'équation de conservation de la quantité de mouvement
R_2	résidu vectoriel pour l'équation de conservation de la masse
S	matrice des contributions relatives aux conditions aux limites de <i>Neumann</i>
t	variable temps
T^K	transformation de l'élément de référence \hat{K} à un élément K
\mathbf{u}	vitesse (vectorielle)
U_k	vecteur vitesse à l'itération k
\mathbf{v}	fonction test pour l'équation de conservation de la quantité de mouvement
W	vecteur de correction en vitesse pour l'algorithme d'Uzawa
\mathbf{x}	variable d'espace
(x, y, z)	point en coordonnées cartésiennes
∂K	frontière de l'élément K
∞	infini

Caractères grecs

$\delta \mathbf{u}$	vecteur correction en vitesse
δU	vecteur des degrés de liberté de correction en vitesse
δp	vecteur correction en pression
δP	vecteur des degrés de liberté de correction en pression
Δp	variation de pression
ϵ	critère d'arrêt

η	viscosité pour un fluide newtonien
η_0	viscosité à cisaillement nul
$\eta(\dot{\gamma}(\mathbf{u}))$	viscosité pour un fluide newtonien généralisé
$\dot{\gamma}(\mathbf{u})$	tenseur du taux de déformation
$ \dot{\gamma}(\mathbf{u}) $	deuxième invariant du tenseur du taux de déformation
Γ	frontière du domaine physique
Γ_1	partie de la frontière du domaine physique
Γ_2	partie de la frontière du domaine physique
$\kappa(A)$	valeur du conditionnement de la matrice A
λ	temps de retard
Ω	domaine physique
$\Psi_{\mathbf{u},j}$	fonction d'interpolation en vitesse
$\psi_{p,j}$	fonction d'interpolation en pression
$\Psi_{\mathbf{u}}$	vecteur des fonctions d'interpolation en vitesse
ψ_p	vecteur des fonctions d'interpolation en pression
ρ	densité
σ	tenseur de Cauchy
(ξ, η, ζ)	points en coordonnées cartésiennes pour le passage à l'élément de référence

Autres symboles

∇	opérateur gradient
$\nabla \cdot$	opérateur divergence
$\ \quad \ $	norme vectorielle

Indices

K	le sur-indice K indique la restriction à l'élément K (élémentaire)
-----	--

Liste des annexes

ANNEXE A	Matrice symétrique et matrice définie positive	54
ANNEXE B	Méthode du gradient conjugué	57
ANNEXE C	Approche informatique	59

Introduction

Depuis ses premières applications, à la fin des années soixante, *la méthode des éléments finis* a connu un essor remarquable et elle est maintenant utilisée dans divers domaines de l'ingénierie tels que la mécanique des solides et la mécanique des fluides.

Ainsi, en mécanique des fluides, par exemple, on a effectué beaucoup de recherches pour tenter de modéliser des écoulements de différents fluides dans différents contextes. On a, entre autres, modélisé par ordinateur l'écoulement de l'air autour des ailes d'avions pour augmenter l'aérodynamisme de celles-ci, tout en évitant les coûts élevés des tests en soufflerie. Toujours en mécanique des fluides, on a également essayé de simuler l'écoulement de fluides newtoniens et non newtoniens dans différents conduits. On parle de fluide newtonien lorsque la viscosité de celui-ci est constante, dans le cas contraire, on parle d'un fluide non newtonien.

Dans la catégorie des fluides non newtoniens, on retrouve les polymères qui sont largement utilisés de nos jours et les technologies qui y sont rattachées sont parfois très complexes. Pour produire bon nombre d'objets que l'on rencontre à tous les jours, il a souvent fallu appliquer des procédés tels que la plastification de granules de plastique, souvent par extrusion, et le moulage par injection pour la mise en forme du polymère. Il est donc utile de pouvoir prédire le comportement de ces fluides afin de bien planifier les installations requises à de tels procédés. Ces polymères fondus sont des fluides dits viscoélastiques et viscoplastiques et leur viscosité diminue en fonction du cisaillement. Il existe plusieurs modèles servant à modéliser la viscosité des polymères (et des fluides non newtoniens en général) et il s'agit de trouver quel modèle s'applique le mieux au fluide dont on veut décrire l'écoulement. Toutefois, une difficulté majeure surgit avec de tels modèles; on obtient des problèmes non linéaires

qu'il est souvent difficile de résoudre. On verra, dans ce mémoire, comment on peut régler ces problèmes de non-linéarité.

Plusieurs écoulements peuvent être modélisés à l'aide des équations de Navier-Stokes ou, dans des cas plus simples, des équations de Stokes. Dans la majorité des cas, comme on ne peut obtenir des solutions exactes à ces équations aux dérivées partielles, on utilise des méthodes numériques comme celle qui fait l'objet de ce mémoire, *la méthode des éléments finis*. Cette méthode conduit, on pourra le constater, à la résolution de systèmes d'équations linéaires de très grande taille. Avec les ordinateurs de plus en plus puissants que l'on dispose de nos jours, il est maintenant possible de résoudre des problèmes d'une telle ampleur, notamment sur des géométries très complexes. Dans les dernières années on a beaucoup travaillé sur les problèmes en 2D (deux dimensions). Il reste du travail à faire en ce qui concerne les problèmes en 3D (trois dimensions). Dans ce mémoire les résultats ont été établis dans la perspective de pouvoir résoudre des problèmes en 2D et en 3D. Cependant, on y présente des stratégies de résolution qui permettent de résoudre des problèmes de grande taille que l'on retrouve généralement dans les problèmes tridimensionnels.

Pour pouvoir tirer le maximum de la mémoire de l'ordinateur, on doit tirer profit de la structure des matrices obtenues lors de la discrétisation. Souvent on ne pourra pas utiliser des méthodes directes telles que la décomposition LU (voir par exemple Fortin (1995)), qui sont très coûteuses en terme de mémoire et il faut plutôt utiliser des méthodes itératives (exemple: gradient conjugué, MINRES ou GMRES pour ne nommer que celles-là). L'article de Carey et al. (1989) présente des méthodes itératives, avec pénalisation, appliquées aux écoulements de fluides newtoniens et newtoniens généralisés. Pour une référence plus générale sur les méthodes itératives, on pourra consulter Greenbaum (1997).

Par la modélisation en mécanique des fluides, on cherche généralement à trouver une solution numérique (lorsque c'est impossible de le faire de façon analytique) pour deux variables: la vitesse et la pression. À ce propos, la méthode d'Uzawa permet de découpler la résolution des systèmes linéaires. On résout d'une part les équations du problème en vitesse et d'autre part celles en pression. Ainsi on peut utiliser toute la mémoire disponible pour résoudre d'abord en vitesse et ensuite utiliser la solution en vitesse pour résoudre le problème en pression. On tire donc profit de la structure de la matrice globale en vitesse-pression et on utilise des solveurs spécialisés à chaque

étape de la résolution. Mentionnons ici les travaux de Robichaud et al. (1990) sur la méthode d'Uzawa pour la résolution des équations de Navier-Stokes.

Dans ce mémoire, on essaie donc de modéliser, via un modèle newtonien généralisé, l'écoulement de fluides newtoniens et non newtoniens à l'aide des équations de Stokes. On y présentera différents modèles pour la viscosité. On discutera de la méthode d'Uzawa ainsi qu'une variante de cette méthode, où on a tenté d'ajouter une pénalisation. Les résultats ont été obtenus à l'aide du code d'éléments finis **MEF++** du *GIREF*, le *Groupe Interdisciplinaire de Recherche en Éléments Finis* de l'Université Laval.

Plusieurs travaux ont été faits concernant les écoulements de fluides newtoniens généralisés. L'article de Gartling (1986) fait une vaste couverture de la méthode des éléments finis appliquée aux écoulements de fluides non newtoniens. Il faut attendre aux années 1980 avant de voir les premiers travaux concernant les écoulements de ces types de fluides sur des géométries tridimensionnelles. Notons l'article de Tanguy et al. (1988), à ce sujet.

Au niveau des applications, on retrouve différents articles décrivant la méthode des éléments finis appliquée à la résolution de problèmes d'écoulement dans des domaines bien précis. Mentionnons les articles de Crochet et al. (1984), qui couvre différentes applications dans le domaine des fluides non-newtoniens, de Bravo et al. (1998), à propos d'écoulements dans des vis d'extrusion et de Bertrand et al. (1999) sur des procédés de mélange.

Chapitre 1

Équations de la physique

1.1 Équations de la mécanique des fluides

Situons-nous dans le contexte de la mécanique des fluides. Nous voulons décrire l'écoulement de divers types de fluides, dans diverses géométries données que nous appellerons Ω et leur frontière Γ . On pourrait penser à n'importe quel type de domaine comme par exemple un tube:

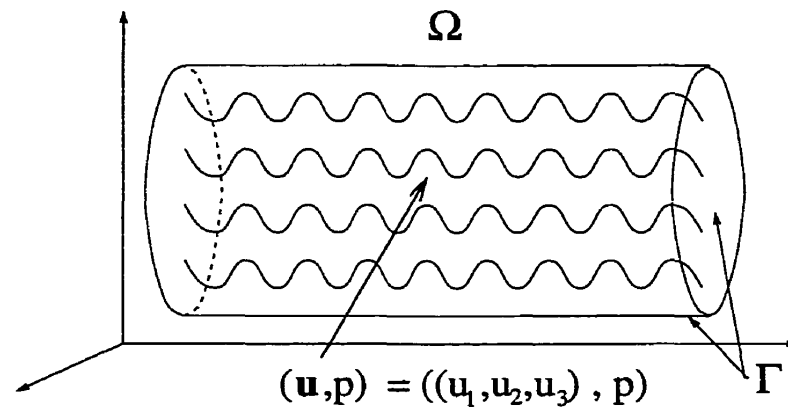


Figure 1.1: Exemple d'un domaine Ω en 3D

On voudra donc décrire le champ de vitesse et la pression à l'intérieur de la géométrie Ω . Un très grand nombre de problèmes de ce genre peuvent être résolus par les équations de Navier-Stokes. On peut se référer à Pironneau (1988) en ce qui concerne la provenance des équations de Navier-Stokes.

Les équations de Navier-Stokes s'écrivent, pour des fluides incompressibles, sous forme tensorielle comme suit :

Équations de Navier-Stokes - Fluides incompressibles

$$\left\{ \begin{array}{l} \text{Conservation de la quantité de mouvement:} \\ \rho \frac{D\mathbf{u}}{Dt} - \nabla \cdot (2\eta \dot{\gamma}(\mathbf{u})) + \nabla p = \rho \mathbf{f} \quad \text{sur } \Omega \\ \\ \text{Conservation de la masse (incompressibilité):} \\ \nabla \cdot \mathbf{u} = 0 \quad \text{sur } \Omega \\ \\ \text{avec des conditions aux limites sur } \Gamma \end{array} \right.$$

Précisons un peu les différentes parties qui constituent les équations de Navier-Stokes. Comme nous l'avons déjà mentionné, les deux variables étudiées dans les problèmes de la mécanique des fluides sont la vitesse et la pression qui sont respectivement représentées par \mathbf{u} , un vecteur de dimension deux ou trois selon que le problème est en deux ou trois dimensions et p un scalaire.

Le premier terme $\rho \frac{D\mathbf{u}}{Dt}$, qui peut se décomposer en $\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u}$, représente le terme d'inertie. Le terme $-\nabla \cdot (2\eta \dot{\gamma}(\mathbf{u}))$ constitue quant à lui le terme visqueux du fluide dans l'écoulement. Le terme η est le coefficient de viscosité, qui peut être constant si le fluide est newtonien ou non constant si le fluide est non newtonien. On y reviendra à la section 1.2, alors qu'on discutera des différents modèles pour la viscosité.

On retrouve également le terme $\dot{\gamma}(\mathbf{u})$, le tenseur du taux de déformation qui s'écrit, dans le cas tridimensionnel, comme suit:

$$\dot{\gamma}(\mathbf{u}) = \frac{1}{2}(\nabla^T \mathbf{u} + \nabla \mathbf{u}) = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{1}{2} \left(\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial u_1}{\partial z} + \frac{\partial u_3}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial u_2}{\partial x} + \frac{\partial u_1}{\partial y} \right) & \frac{\partial u_2}{\partial y} & \frac{1}{2} \left(\frac{\partial u_2}{\partial z} + \frac{\partial u_3}{\partial y} \right) \\ \frac{1}{2} \left(\frac{\partial u_3}{\partial x} + \frac{\partial u_1}{\partial z} \right) & \frac{1}{2} \left(\frac{\partial u_3}{\partial y} + \frac{\partial u_2}{\partial z} \right) & \frac{\partial u_3}{\partial z} \end{bmatrix} \quad (1.1)$$

Pour que le problème soit bien posé, on se doit d'imposer le comportement du fluide, soit la vitesse ou encore la pression, sur la frontière Γ (ou sur une partie de

celle-ci), à l'aide des conditions aux limites. On impose généralement $\mathbf{u} = 0$ sur les parois solides de Γ ; puisque le fluide adhère aux parois, la vitesse y est nulle. On peut imposer une vitesse quelconque sur une partie de la frontière, disons Γ_1 via les conditions aux limites de type *Dirichlet*. On le verra, il est également possible d'imposer le terme de pression $(-pI + 2\eta\dot{\gamma}(\mathbf{u})) \cdot \mathbf{n}$ sur une partie de la frontière, Γ_2 , via les conditions aux limites de type *Neumann*. Il est aussi possible de combiner les deux types de conditions, sans toutefois imposer les deux types de conditions sur une même partie de la frontière.

Les équations de Navier-Stokes sont très générales et on peut résoudre un tas de problèmes avec celles-ci. Il est cependant possible de les simplifier dans certains cas particuliers. Dans le cas d'écoulements lents par exemple, on peut négliger le terme d'inertie $\rho \frac{D\mathbf{u}}{Dt}$ par rapport aux autres termes des équations de conservation de la quantité de mouvement des équations de Navier-Stokes.

Avec cette simplification, on arrive aux équations de Stokes pour les fluides incompressibles.

Équations de Stokes - Fluides incompressibles

$$\left\{ \begin{array}{l} \text{Conservation de la quantité de mouvement:} \\ \quad -\nabla \cdot (2\eta \dot{\gamma}(\mathbf{u})) + \nabla p = \mathbf{f} \quad \text{sur } \Omega \\ \\ \text{Conservation de la masse (incompressibilité):} \\ \quad \nabla \cdot \mathbf{u} = 0 \quad \text{sur } \Omega \\ \\ \text{Conditions aux limites:} \\ \quad \left\{ \begin{array}{ll} \text{Dirichlet:} & \mathbf{u} = \mathbf{g} \quad \text{sur } \Gamma_1 \\ \text{Neumann:} & (-pI + 2\eta\dot{\gamma}(\mathbf{u})) \cdot \mathbf{n} = \mathbf{h} \quad \text{sur } \Gamma_2 \end{array} \right. \end{array} \right.$$

où \mathbf{g} et \mathbf{h} sont des fonctions connues sur la frontière de Ω , I est le tenseur identité et \mathbf{n} est un vecteur normal extérieur à Γ .

Dans ce mémoire, nous allons nous attarder à ces équations de Stokes et nous allons montrer comment les résoudre à l'aide de méthodes numériques.

1.2 Modèles pour la viscosité

Lorsqu'on applique une contrainte de cisaillement à un fluide quelconque, il en résulte une déformation de celui-ci. La viscosité η d'un fluide est en quelque sorte la capacité de ce dernier à résister plus ou moins à cette déformation. Pour les fluides newtoniens, on pense ici à de l'eau ou de l'air, la viscosité est constante. Ainsi la déformation est proportionnelle à la contrainte qu'on applique au fluide. Dans les équations de Stokes (ou plus généralement dans celles de Navier-Stokes), on posera alors tout simplement $\eta = \eta_0$, une constante.

En revanche, les fluides non newtoniens tels que le sang, le pétrole et les polymères fondus possèdent une viscosité non constante qui diminue généralement lorsque le taux de cisaillement augmente. On remarque qu'à l'échelle logarithmique, la viscosité décroît de façon linéaire en fonction du cisaillement (voir figure 1.2 à la page 8). Il faut donc se doter de modèles rhéologiques pour définir la viscosité en fonction du tenseur $\dot{\gamma}(\mathbf{u})$, ou plus précisément en fonction de $|\dot{\gamma}(\mathbf{u})|$, le deuxième invariant du tenseur du taux de déformation, défini par:

$$|\dot{\gamma}(\mathbf{u})|^2 = 2 \dot{\gamma}(\mathbf{u}) : \dot{\gamma}(\mathbf{u}) = 2 \sum_{ij} (\dot{\gamma}(\mathbf{u}))_{ij} (\dot{\gamma}(\mathbf{u}))_{ji}$$

Plusieurs modèles qui ont été proposés ont donc la forme suivante $\eta = \eta(|\dot{\gamma}(\mathbf{u})|)$. Ce sont les modèles pour les fluides dits newtoniens généralisés.

Dans le cas des polymères fondus, des fluides aussi non newtoniens, on observe la présence de ce qu'on appelle le plateau newtonien. Ce plateau est présent à de faibles taux de cisaillement. Pour des cisaillements plus élevés, on retrouve un comportement semblable à ceux des autres fluides non newtoniens. On le verra, certains modèles tiennent compte de ce phénomène alors que d'autres n'y arrivent pas.

Sur la figure 1.2, de la page suivante, on présente les différentes situations possibles en échelle logarithmique. La courbe (qui est une droite) horizontale représente le cas de fluides newtoniens alors que les deux autres courbes représentent le cas de fluides non newtoniens, qu'on appelle aussi fluides newtoniens généralisés. On présente aussi le plateau newtonien.

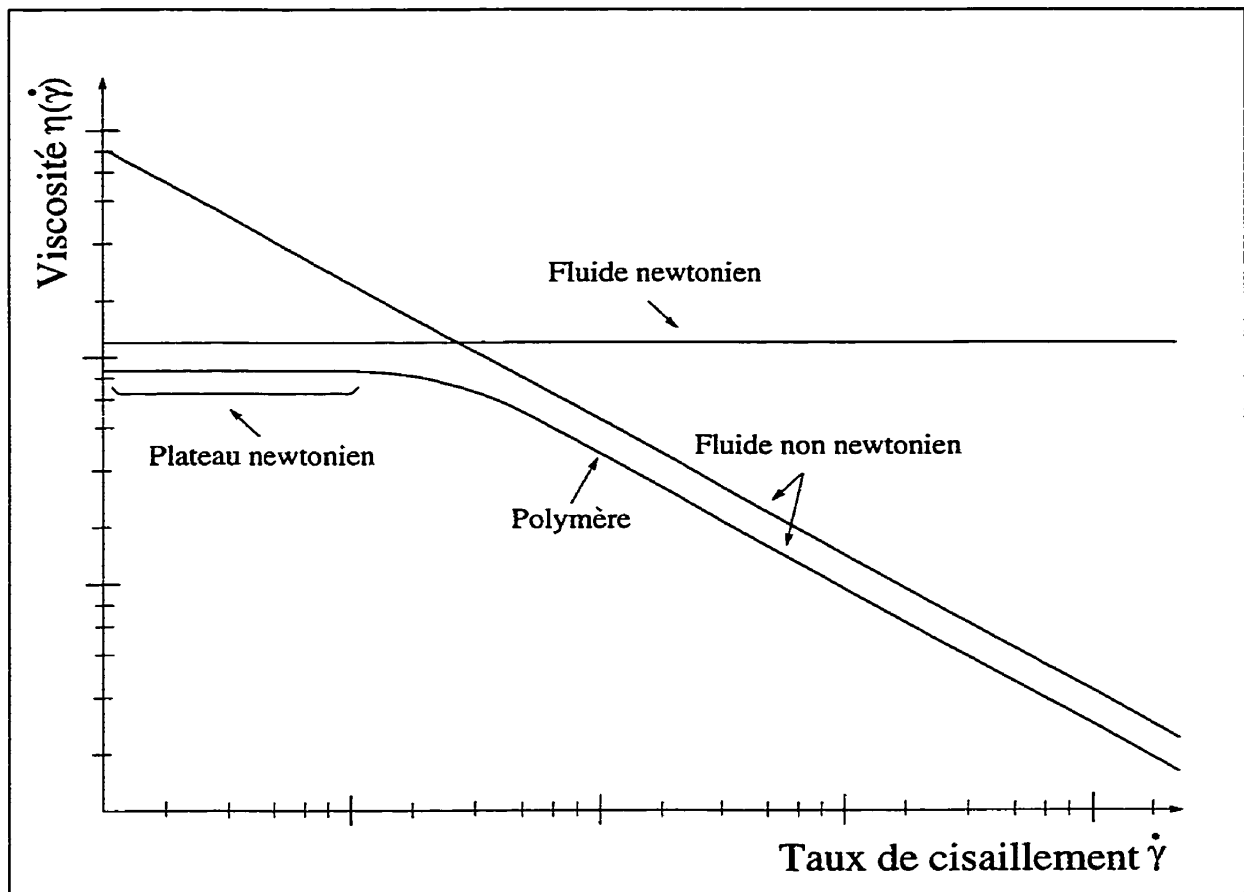


Figure 1.2: Viscosité en fonction du cisaillement pour différents fluides

Un premier modèle pour la viscosité des fluides viscoplastiques est celui *d'Ostwald-De Waele*, aussi connu sous le nom de *loi de puissance*.

Modèle *loi de puissance*

$$\eta(|\dot{\gamma}(\mathbf{u})|) = \eta_0 |\dot{\gamma}(\mathbf{u})|^{N-1}$$

Bien que ce modèle s'applique à plusieurs fluides non newtoniens dont les métaux à l'état fondu, il ne permet pas de simuler la présence des plateaux newtoniens dont nous avons discuté auparavant. Il n'en demeure pas moins que ce modèle est largement utilisé.

Pour pallier à ce défaut du modèle d'*Ostwald-De Waele*, on peut utiliser le modèle de *Carreau*.

Modèle de *Carreau*

$$\eta(|\dot{\gamma}(\mathbf{u})|) = \eta_0 (1 + \lambda^2 |\dot{\gamma}(\mathbf{u})|^2)^{\frac{N-1}{2}}$$

où les différents paramètres des deux modèles précédents sont donnés par:

- η_0 : La viscosité à cisaillement nul
- λ : Un temps de retard
- N : L'indice de pseudoplasticité ($0 \leq N \leq 1$)

On peut généraliser l'écriture de ces différents modèles à l'aide du modèle suivant:

$$\eta(|\dot{\gamma}(\mathbf{u})|) = \eta_0 (C + \lambda^2 |\dot{\gamma}(\mathbf{u})|^2)^{\frac{N-1}{2}} \quad (1.2)$$

où: η_0 , λ et N ont été définis avec le modèle de Carreau.

Ainsi on retrouve les différents modèles présentés en fixant les différents paramètres de façon appropriée:

- Modèle pour un fluide newtonien: $N = 1$
- Modèle *loi de puissance*: $C=0$ et $\lambda = 1$
- Modèle de *Carreau*: $C=1$

Pour calculer les paramètres η_0 , λ et N dans les modèles ci-haut, on procède de façon empirique. On mesure, pour un fluide donné, un certain nombre de valeurs de viscosité en fonction du taux de cisaillement à l'aide d'un rhéomètre. On applique une méthode de moindres carrés afin d'obtenir les constantes recherchées pour le fluide en question. (On retrouve dans Fortin (1995) deux exemples concrets de ce procédé pour les modèles de *loi de puissance* et de *Carreau*).

D'autres modèles peuvent aussi être utilisés, notamment les modèles de *Ellis*, de *Cross*, de *De Kee* et de *Carreau-Yasuda* pour ne nommer que ceux-ci. On peut consulter Carreau et al. (1997) pour une description complète de ces différents modèles.

On introduit donc ce modèle newtonien généralisé dans les équations de Stokes, pour obtenir:

Équations de Stokes - Fluides newtoniens généralisés, incompressibles

$$\left\{ \begin{array}{l} \text{Conservation de la quantité de mouvement:} \\ \quad -\nabla \cdot \{ 2 \eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u}) \} + \nabla p = \mathbf{f} \quad \text{sur } \Omega \\ \\ \text{Conservation de la masse (incompressibilité):} \\ \quad \nabla \cdot \mathbf{u} = 0 \quad \text{sur } \Omega \\ \\ \text{Conditions aux limites:} \\ \quad \left\{ \begin{array}{ll} \text{Dirichlet:} & \mathbf{u} = \mathbf{g} \quad \text{sur } \Gamma_1 \\ \text{Neumann:} & (-pI + 2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u})) \cdot \mathbf{n} = \mathbf{h} \quad \text{sur } \Gamma_2 \end{array} \right. \end{array} \right.$$

où: $\eta(|\dot{\gamma}(\mathbf{u})|) = \eta_0 (C + \lambda^2 |\dot{\gamma}(\mathbf{u})|^2)^{\frac{N-1}{2}}$

Notons que si on veut imposer la vitesse, on utilise les conditions aux limites de *Dirichlet*. L'imposition de la pression se fait, quant à elle, via l'imposition de

$$(-pI + 2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u})) \cdot \mathbf{n} \equiv \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h}$$

où $\boldsymbol{\sigma}$ est connu sous le nom de *tenseur de Cauchy*.

En général, on ne peut pas résoudre les équations de Stokes, présentés ci-haut, de façon analytique, il faut donc utiliser des méthodes numériques. Nous allons voir au prochain chapitre comment on peut résoudre ces équations à l'aide de la méthode des éléments finis.

Chapitre 2

Méthode des éléments finis

La méthode des éléments finis (voir par exemple Reddy (1993)) est un outil très puissant de discrétisation pour la résolution numérique d'équations aux dérivées partielles, telles que les équations de Stokes. Comme son nom l'indique, cette méthode consiste à découper le domaine Ω en un nombre finis d'éléments K , le plus souvent des triangles en 2D et des tétraèdres en 3D (voir figure 2.1). Bien que cette étape soit cruciale, nous n'entrerons pas plus dans le sujet complexe (surtout en 3D) de la fabrication de maillages. Il existe plusieurs logiciels pour créer de tels maillages en 2D et en 3D.

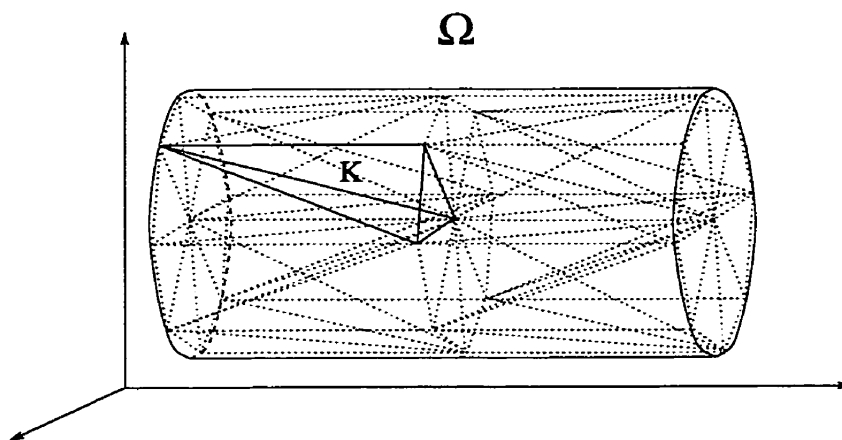


Figure 2.1: Exemple d'un maillage du domaine Ω en 3D

Ensuite on construit des équations discrètes qui approchent les équations à résoudre. Les inconnues de ces équations discrètes sont les valeurs de la solution sur les noeuds

de calcul (souvent les sommets des éléments ou encore le milieu des arêtes).

On obtient donc un grand système d'équations linéaires et parfois non linéaires qu'il faut résoudre à l'aide de méthodes d'analyse numérique. Une fois le système résolu, on obtient la solution seulement aux noeuds de calcul des éléments. Ainsi plus le maillage contient d'éléments, plus on obtient une solution précise.

2.1 Formulation variationnelle des équations de Stokes

Comme nous l'avons déjà mentionné, l'une des étapes fondamentales du processus de résolution consiste à construire les équations discrètes à partir des équations continues à résoudre, dans notre cas les équations de Stokes. Pour ce faire, on se donne la formulation variationnelle (ou formulation faible) des équations de Stokes.

Pour éviter d'alourdir le texte, nous ne considérerons, dans un premier temps, que le cas de conditions aux limites de type *Dirichlet*. De plus, on supposera que l'on a des conditions limites homogènes, c'est-à-dire que $\mathbf{u}=0$ sur Γ . Dans le cas de conditions aux limites non homogènes, on devrait procéder à un relèvement des conditions limites, voir Fortin et Garon (2000).

Nous allons tout d'abord définir les espaces fonctionnels nécessaires à l'application de la méthode des éléments finis. Étant donné que l'on considère seulement le cas des conditions aux limites de type *Dirichlet* homogènes, nous aurons besoin des espaces fonctionnels suivants.

$$\begin{aligned} V &= (H_0^1(\Omega))^d \quad (d = 2 \text{ ou } 3 \text{ selon la dimension de } \Omega) \\ \text{et} \\ Q &= L^2(\Omega) \end{aligned}$$

où:

$$\begin{aligned} L^2(\Omega) &= \{q : \Omega \rightarrow \mathbf{R} \mid \int_{\Omega} (q(\mathbf{x}))^2 d\mathbf{x} < \infty\} \\ H_0^1(\Omega) &= \{u \in L^2(\Omega) \mid \frac{\partial u}{\partial x_i} \in L^2(\Omega) \text{ avec } u = 0 \text{ sur } \Gamma\} \end{aligned}$$

Les conditions aux limites de type *Neumann* se traiteraient de façon assez similaire, avec un travail supplémentaire, en choisissant cependant les espaces fonctionnels appropriés (voir Fortin et Garon (2000)).

On multiplie donc l'équation de conservation de la quantité de mouvement par une fonction dite test $\mathbf{v} \in V$ et l'équation de la conservation de la masse par une autre fonction test $q \in Q$ puis on intègre sur Ω :

$$\begin{aligned} \int_{\Omega} (-\nabla \cdot \{2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u})\} \cdot \mathbf{v} + \nabla p \cdot \mathbf{v}) d\mathbf{x} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \\ \int_{\Omega} q \nabla \cdot \mathbf{u} d\mathbf{x} &= 0 \end{aligned} \quad (2.1)$$

Si on utilise le théorème de la divergence, l'équation 2.1 devient:

$$\begin{aligned} \int_{\Omega} (2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u}) : \nabla \mathbf{v}) d\mathbf{x} - \int_{\Gamma} (2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u}) \cdot \mathbf{n}) \mathbf{v} ds - \\ \int_{\Omega} p \nabla \cdot \mathbf{v} d\mathbf{x} + \int_{\Gamma} (p \mathbf{n}) \cdot \mathbf{v} ds = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \end{aligned}$$

c'est-à-dire:

$$\begin{aligned} \int_{\Omega} (2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u}) : \nabla \mathbf{v}) d\mathbf{x} - \int_{\Omega} p \nabla \cdot \mathbf{v} d\mathbf{x} - \\ \int_{\Gamma} \{(-p\mathbf{I} + 2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u})) \cdot \mathbf{n}\} \cdot \mathbf{v} ds = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \end{aligned}$$

\mathbf{n} étant le vecteur unitaire normal extérieur à Γ .

Notons qu'en intégrant à l'aide du théorème de la divergence, nous avons fait apparaître le terme $(-p\mathbf{I} + 2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u})) \cdot \mathbf{n}$ relatif aux conditions aux limites de *Neumann* que nous avons vues au chapitre 1. Ce terme est aussi souvent appelé la condition naturelle du problème de Stokes.

Étant donné que nous considérons le cas des conditions aux limites de *Dirichlet* homogènes, on aura $\mathbf{v}|_{\Gamma} = 0$, on obtient donc la formulation variationnelle du problème de Stokes de la page suivante.

Formulation variationnelle (Conditions limites de Dirichlet)

On cherche $\mathbf{u} \in V$ et $p \in Q$ tels que:

$$\begin{cases} \int_{\Omega} (2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u}) : \dot{\gamma}(\mathbf{v})) d\mathbf{x} - \int_{\Omega} p \nabla \cdot \mathbf{v} d\mathbf{x} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} & \forall \mathbf{v} \in V \\ \int_{\Omega} q \nabla \cdot \mathbf{u} d\mathbf{x} = 0 & \forall q \in Q \end{cases}$$

2.2 Linéarisation (méthode de Newton)

La formulation variationnelle telle que présentée ci-haut est non linéaire en raison du terme $\eta(|\dot{\gamma}(\mathbf{u})|)$, qui, dans le cas d'un fluide non newtonien, dépend de \mathbf{u} (voir l'équation 1.2) . Il faut procéder à une linéarisation de la formulation. Pour ce faire, on utilise une méthode classique de l'analyse numérique, la méthode de Newton (voir par exemple Fortin (1995)).

On pose:

$$\begin{aligned} \mathcal{R}_1((\mathbf{u}, p), \mathbf{v}) &= \int_{\Omega} 2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u}) : \dot{\gamma}(\mathbf{v}) d\mathbf{x} \\ &\quad - \int_{\Omega} p \nabla \cdot \mathbf{v} d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \end{aligned} \quad (2.2)$$

$$\mathcal{R}_2((\mathbf{u}, p), q) = \int_{\Omega} q \nabla \cdot \mathbf{u} d\mathbf{x} \quad (2.3)$$

Le problème non linéaire précédent devient alors équivalent à celui de:

Trouver $(\mathbf{u}, p) \in V \times Q$ tels que

$$\begin{cases} \mathcal{R}_1((\mathbf{u}, p), \mathbf{v}) = 0 \\ \mathcal{R}_2((\mathbf{u}, p), q) = 0 \end{cases}$$

pour tout $\mathbf{v} \in V$ et $q \in Q$

Pour ce faire, on se donne des solutions initiales \mathbf{u}_0 et p_0 , puis on cherche des corrections appropriées $\delta \mathbf{u}$ et δp telles que:

$$\begin{cases} \mathcal{R}_1((\mathbf{u}_0 + \delta \mathbf{u}, p_0 + \delta p), \mathbf{v}) = 0 \\ \mathcal{R}_2((\mathbf{u}_0 + \delta \mathbf{u}, p_0 + \delta p), q) = 0 \end{cases}$$

Ce qui, en utilisant le développement de Taylor, est équivalent à:

$$\begin{cases} \mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v}) + \frac{\partial \mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v})}{\partial \mathbf{u}} \cdot \delta \mathbf{u} + \frac{\partial \mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v})}{\partial p} \cdot \delta p + \dots = 0 \\ \mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v}) + \frac{\partial \mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v})}{\partial \mathbf{u}} \cdot \delta \mathbf{u} + \frac{\partial \mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v})}{\partial p} \cdot \delta p + \dots = 0 \end{cases}$$

En négligeant les termes d'ordre supérieur ou égal à 2, on obtient le système linéaire suivant:

$$\begin{cases} \frac{\partial \mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v})}{\partial \mathbf{u}} \cdot \delta \mathbf{u} + \frac{\partial \mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v})}{\partial p} \cdot \delta p = -\mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v}) \\ \frac{\partial \mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v})}{\partial \mathbf{u}} \cdot \delta \mathbf{u} + \frac{\partial \mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v})}{\partial p} \cdot \delta p = -\mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v}) \end{cases}$$

On utilise la notion de dérivée d'une fonctionnelle (ou dérivée de Gâteaux), ainsi que la définition de $\eta(|\dot{\gamma}(\mathbf{u})|)$ (voir l'équation 1.2) pour calculer les quatre dérivées présentes dans le système précédent:

$$\begin{aligned} \bullet \quad & \frac{\partial \mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v})}{\partial \mathbf{u}} \cdot \delta \mathbf{u} = \left. \frac{d \mathcal{R}_1((\mathbf{u}_0 + \xi \delta \mathbf{u}, p_0), \mathbf{v})}{d \xi} \right|_{\xi=0} \\ & = \frac{d}{d \xi} \left\{ \int_{\Omega} 2 \eta_0 \{ C + \lambda^2 |\dot{\gamma}(\mathbf{u}_0 + \xi \delta \mathbf{u})|^2 \}^{\frac{N-1}{2}} \dot{\gamma}(\mathbf{u}_0 + \xi \delta \mathbf{u}) : \dot{\gamma}(\mathbf{v}) d\mathbf{x} - \right. \\ & \quad \left. \int_{\Omega} p \nabla \cdot \mathbf{v} d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \right\} \Big|_{\xi=0} \\ & = \frac{d}{d \xi} \int_{\Omega} 2 \eta_0 \{ C + \lambda^2 (|\dot{\gamma}(\mathbf{u}_0)|^2 + 4 \xi \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\delta \mathbf{u}) + \xi^2 |\dot{\gamma}(\delta \mathbf{u})|^2) \}^{\frac{N-1}{2}} \dots \\ & \quad \dots \{ \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\mathbf{v}) + \xi \dot{\gamma}(\delta \mathbf{u}) : \dot{\gamma}(\mathbf{v}) \} d\mathbf{x} \Big|_{\xi=0} \end{aligned}$$

$$\begin{aligned}
&= \int_{\Omega} 2\eta_0 \left[\frac{N-1}{2} \{C + \lambda^2(|\dot{\gamma}(\mathbf{u}_0)|^2 + 4\xi \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\delta\mathbf{u}) + \xi^2 |\dot{\gamma}(\delta\mathbf{u})|^2)\}^{\frac{N-1}{2}-1} \dots \right. \\
&\quad \dots \{4\lambda^2 \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\delta\mathbf{u}) + 2\xi |\dot{\gamma}(\delta\mathbf{u})|^2\} \{ \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\mathbf{v}) + \xi \dot{\gamma}(\delta\mathbf{u}) : \dot{\gamma}(\mathbf{v}) \} + \\
&\quad \left. \{C + \lambda^2(|\dot{\gamma}(\mathbf{u}_0)|^2 + 4\xi \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\delta\mathbf{u}) + \xi^2 |\dot{\gamma}(\delta\mathbf{u})|^2)\}^{\frac{N-1}{2}} \{ \dot{\gamma}(\delta\mathbf{u}) : \dot{\gamma}(\mathbf{v}) \} \right] d\mathbf{x} \Big|_{\xi=0} \\
&= \int_{\Omega} 4(N-1)\lambda^2\eta_0 \{C + \lambda^2(|\dot{\gamma}(\mathbf{u}_0)|^2)\}^{\frac{N-3}{2}} \{ \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\delta\mathbf{u}) \} \{ \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\mathbf{v}) \} d\mathbf{x} + \\
&\quad \int_{\Omega} 2\eta_0 \{C + \lambda^2(|\dot{\gamma}(\mathbf{u}_0)|^2)\}^{\frac{N-1}{2}} \{ \dot{\gamma}(\delta\mathbf{u}) : \dot{\gamma}(\mathbf{v}) \} d\mathbf{x}
\end{aligned}$$

$$\begin{aligned}
\bullet \quad &\frac{\partial \mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v})}{\partial p} \cdot \delta p = \frac{d \mathcal{R}_1((\mathbf{u}_0, p_0 + \xi \delta p), \mathbf{v})}{d\xi} \Big|_{\xi=0} \\
&= \frac{d}{d\xi} \left\{ \int_{\Omega} \eta(|\dot{\gamma}(\mathbf{u}_0)|) \dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\mathbf{v}) d\mathbf{x} - \int_{\Omega} (p_0 + \xi \delta p) \nabla \cdot \mathbf{v} d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \right\} \Big|_{\xi=0} \\
&= - \int_{\Omega} \delta p \nabla \cdot \mathbf{v} d\mathbf{x}
\end{aligned}$$

$$\begin{aligned}
\bullet \quad &\frac{\partial \mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v})}{\partial \mathbf{u}} \cdot \delta \mathbf{u} = \frac{d \mathcal{R}_2((\mathbf{u}_0 + \xi \delta \mathbf{u}, p_0), \mathbf{v})}{d\xi} \Big|_{\xi=0} \\
&= \frac{d}{d\xi} \int_{\Omega} q \nabla \cdot (\mathbf{u}_0 + \xi \delta \mathbf{u}) d\mathbf{x} \Big|_{\xi=0} \\
&= \int_{\Omega} q \nabla \cdot \delta \mathbf{u} d\mathbf{x}
\end{aligned}$$

$$\begin{aligned}
\bullet \quad &\frac{\partial \mathcal{R}_2((\mathbf{u}_0, p_0), \mathbf{v})}{\partial p} \cdot \delta p = \frac{d \mathcal{R}_2((\mathbf{u}_0, p_0 + \xi \delta p), \mathbf{v})}{d\xi} \Big|_{\xi=0} \\
&= \frac{d}{d\xi} \int_{\Omega} q \nabla \cdot \mathbf{u}_0 d\mathbf{x} \Big|_{\xi=0} \\
&= 0
\end{aligned}$$

On introduit le résultat des calculs des quatre dérivées dans le système linéaire précédent, pour obtenir la formulation variationnelle linéarisée du problème de Stokes non linéaire.

Formulation variationnelle linéarisée

On cherche $\delta \mathbf{u} \in V$ et $\delta p \in Q$ tels que:

$$\left\{ \begin{array}{l} \int_{\Omega} \left\{ 4(N-1) \lambda^2 \eta_0 \{C + \lambda^2 |\dot{\gamma}(\mathbf{u}_0)|^2\}^{\frac{N-3}{2}} \{\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\delta \mathbf{u})\} \{\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\mathbf{v})\} \right\} d\mathbf{x} + \\ \int_{\Omega} \left\{ 2\eta(|\dot{\gamma}(\mathbf{u}_0)|) (\dot{\gamma}(\delta \mathbf{u}) : \dot{\gamma}(\mathbf{v})) \right\} d\mathbf{x} - \int_{\Omega} \delta p \nabla \cdot \mathbf{v} d\mathbf{x} = -\mathcal{R}_1((\mathbf{u}_0, p_0), \mathbf{v}) \\ \int_{\Omega} q \nabla \cdot \delta \mathbf{u} d\mathbf{x} = -\mathcal{R}_2((\mathbf{u}_0, p_0), q) \end{array} \right.$$

$$\forall \mathbf{v} \in V \text{ et } \forall q \in Q$$

Étant donné qu'on utilise la méthode de Newton, on commet une erreur de troncature. Il faut donc itérer pour converger vers la solution. On passe d'une itération à la suivante en posant:

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \delta \mathbf{u} \quad \text{et} \quad p_k = p_{k-1} + \delta p$$

jusqu'à la convergence de la solution en vitesse et en pression.

La prochaine étape, qui est le fondement de la méthode des éléments finis, sera de construire le problème discret relié au problème de Stokes.

2.3 Discrétisation du problème de Stokes

Pour procéder à la discrétisation, on doit se donner la formulation variationnelle élémentaire, c'est-à-dire la formulation variationnelle linéarisée présentée ci-haut mais sur chaque élément. Pour ce faire, on n'intègre plus sur Ω mais bien sur chacun des éléments K . En utilisant la définition de $\mathcal{R}_1((\mathbf{u}, p), \mathbf{v})$ et de $\mathcal{R}_2((\mathbf{u}, p), q)$ (voir (2.2)-(2.3)), en se rappelant que $(-pI + 2\eta(|\dot{\gamma}(\mathbf{u})|)\dot{\gamma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{h}$ sur ∂K et en introduisant \mathbf{h}^K et \mathbf{f}^K , respectivement la restriction de \mathbf{h} et \mathbf{f} à l'élément K , on obtient la formulation variationnelle de la page suivante:

Formulation variationnelle linéarisée élémentaire

Chercher $\delta \mathbf{u}$ et δp tels que:

$$\left\{ \begin{array}{l} \int_K \left\{ 4(N-1) \lambda^2 \eta_0 \{C + \lambda^2 |\dot{\gamma}(\mathbf{u}_0)|^2\}^{\frac{N-3}{2}} \{\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\delta \mathbf{u})\} \{\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\mathbf{v})\} \right\} d\mathbf{x} + \\ \int_K \left\{ 2\eta(|\dot{\gamma}(\mathbf{u}_0)|) (\dot{\gamma}(\delta \mathbf{u}) : \dot{\gamma}(\mathbf{v})) \right\} d\mathbf{x} - \int_K \delta p \nabla \cdot \mathbf{v} d\mathbf{x} = \int_{\partial K} \mathbf{h}^K \cdot \mathbf{v} ds - \\ \int_K 2\eta(|\dot{\gamma}(\mathbf{u}_0)|) (\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\mathbf{v})) d\mathbf{x} + \int_K p_0 \nabla \cdot \mathbf{v} d\mathbf{x} + \int_K \mathbf{f}^K \cdot \mathbf{v} d\mathbf{x} \\ - \int_K q \nabla \cdot \delta \mathbf{u} d\mathbf{x} = \int_K q \nabla \cdot \mathbf{u}_0 d\mathbf{x} \end{array} \right.$$

$$\forall \mathbf{v} \in V \text{ et } \forall q \in Q$$

Nous utilisons ensuite des fonctions d'interpolation $\Psi_{\mathbf{u},j}^K(\mathbf{x})$ et $\psi_{p,j}^K(\mathbf{x})$ pour approximer respectivement $\delta \mathbf{u}$ et δp . On reviendra sur le choix de ces fonctions d'interpolation un peu plus loin, après avoir discuté du choix du type d'éléments qu'on utilisera. On posera:

$$\delta \mathbf{u}|_K \simeq \delta \mathbf{u}^K = \sum_{j=1}^{n_D^{uK}} \delta U_j^K \Psi_{\mathbf{u},j}^K$$

$$\delta p|_K \simeq \delta p^K = \sum_{j=1}^{n_D^{pK}} \delta P_j^K \psi_{p,j}^K$$

où n_D^{uK} et n_D^{pK} dénotent le nombre de degrés de liberté en vitesse et en pression de l'élément K . On prendra ensuite successivement $\mathbf{v} = \Psi_{\mathbf{u},i}^K, i=1, \dots, n_D^{uK}$ et $q = \psi_{p,i}^K, i=1, \dots, n_D^{pK}$. Ce processus nous conduit à un système de plusieurs équations en termes des inconnues δU_m^K ($m = 1, \dots, n_D^{uK}$) et δP_n^K ($n = 1, \dots, n_D^{pK}$).

Plus précisément, on obtiendra le système d'équations élémentaire suivant:

$$\begin{bmatrix} (\hat{A}^K + \tilde{A}^K) & (B^K)^T \\ B^K & 0 \end{bmatrix} \begin{bmatrix} \delta U^K \\ \delta P^K \end{bmatrix} = \begin{bmatrix} -\tilde{A}^K U_0^K - (B^K)^T P_0^K \\ -B^K U_0^K \end{bmatrix} + \begin{bmatrix} F^K \\ 0 \end{bmatrix} + \begin{bmatrix} S^K \\ 0 \end{bmatrix} \quad (2.4)$$

où on note le vecteur δU^K contenant les n_D^{uK} degrés de liberté en vitesse de l'élément

K de la façon suivante:

$$\delta U^K = \begin{bmatrix} \delta U_1^K \\ \delta U_2^K \\ \delta U_3^K \end{bmatrix} = \begin{bmatrix} \delta u_{1,1}^K \\ \delta u_{1,2}^K \\ \vdots \\ \delta u_{1,n_c^u}^K \\ \hline \delta u_{2,1}^K \\ \delta u_{2,2}^K \\ \vdots \\ \delta u_{2,n_c^u}^K \\ \hline \delta u_{3,1}^K \\ \delta u_{3,2}^K \\ \vdots \\ \delta u_{3,n_c^u}^K \end{bmatrix}$$

Le sous-indice k des δU_k^K , se rapporte à chacune des composantes (x,y ou z). L'indice n_c^u , quant à lui, dénote le nombre de noeuds de calcul en vitesse sur chacun des éléments.

Pour compléter cette notation, on utilisera les fonctions d'interpolation vectorielles $\Psi_{u,j}^K$ exprimées sous la forme:

$$\Psi_u^K = \begin{bmatrix} (\psi_{u,1}^K, 0, 0) \\ (\psi_{u,2}^K, 0, 0) \\ \dots \\ (\psi_{u,n_c^u}^K, 0, 0) \\ \hline (0, \psi_{u,1}^K, 0) \\ (0, \psi_{u,2}^K, 0) \\ \dots \\ (0, \psi_{u,n_c^u}^K, 0) \\ \hline (0, 0, \psi_{u,1}^K) \\ (0, 0, \psi_{u,2}^K) \\ \dots \\ (0, 0, \psi_{u,n_c^u}^K) \end{bmatrix}$$

Pour ce qui est de la notation en pression, c'est beaucoup plus simple étant donné que la pression est un scalaire. On définit le vecteur contenant tous les degrés de liberté en pression de l'élément K . Comme la pression n'a qu'une composante, il y a autant de degré de liberté en pression sur un élément qu'il y a de noeuds de calcul. On notera donc le vecteur δP^K comme suit:

$$\delta P^K = \begin{bmatrix} \delta p_1^K \\ \delta p_2^K \\ \vdots \\ \delta p_{n_c^p}^K \end{bmatrix}$$

et on prendra les fonctions d'interpolation suivantes:

$$\psi_p^K = \begin{bmatrix} \psi_{p,1}^K \\ \psi_{p,2}^K \\ \dots \\ \psi_{p,n_c^p}^K \end{bmatrix}$$

L'indice n_c^p , dénotant le nombre de noeuds de calcul en pression sur chacun des éléments.

Les différents termes du système 2.4 sont donnés par:

$$(\widehat{A}^K)_{ij} = \int_K 4(N-1) \lambda^2 \eta_0 \{C + \lambda^2 |\dot{\gamma}(\mathbf{u}_0)|^2\}^{\frac{N-3}{2}} \{\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\Psi_{\mathbf{u},j}^K)\} \{\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\Psi_{\mathbf{u},i}^K)\} d\mathbf{x}$$

$$(\widetilde{A}^K)_{ij} = \int_K \left\{ 2 \eta(|\dot{\gamma}(\mathbf{u}_0)|) (\dot{\gamma}(\Psi_{\mathbf{u},j}^K) : \dot{\gamma}(\Psi_{\mathbf{u},i}^K)) \right\} d\mathbf{x}$$

$$(B^K)_{ij} = - \int_K \psi_{p,i}^K \nabla \cdot \psi_{p,j}^K d\mathbf{x} = ((B^K)^T)_{ji}$$

$$(S^K)_i = \int_{\partial K} h^K \cdot \Psi_{\mathbf{u},i}^K d\mathbf{x}$$

$$(\widetilde{A}^K U_0^K)_i = \int_K \left\{ 2 \eta(|\dot{\gamma}(\mathbf{u}_0)|) (\dot{\gamma}(\mathbf{u}_0) : \dot{\gamma}(\Psi_{\mathbf{u},i}^K)) \right\} d\mathbf{x}$$

$$((B^K)^T P_0^K)_i = - \int_K p_0 \nabla \cdot \Psi_{u,i}^K$$

$$(F^K)_i = \int_K f^K \cdot \Psi_{u,i}^K d\mathbf{x}$$

$$(B^K U_0^K)_i = - \int_K \psi_{p,i}^K \nabla \cdot \mathbf{u}_0 d\mathbf{x}$$

Il est à noter qu'on effectue le calcul des intégrales précédentes sur chacun des éléments via un changement de variables pour permettre de faire tous les calculs sur un élément de référence \hat{K} (voir figures (2.2)-(2.3)). On se donne donc une transformation $T^K : \hat{K} \rightarrow K$ qui envoie l'élément de référence \hat{K} sur l'élément K .

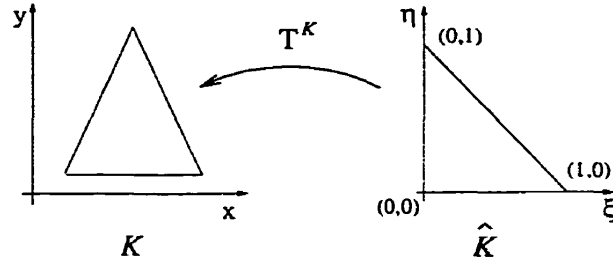


Figure 2.2: Passage d'un élément K à l'élément de référence \hat{K} en 2D

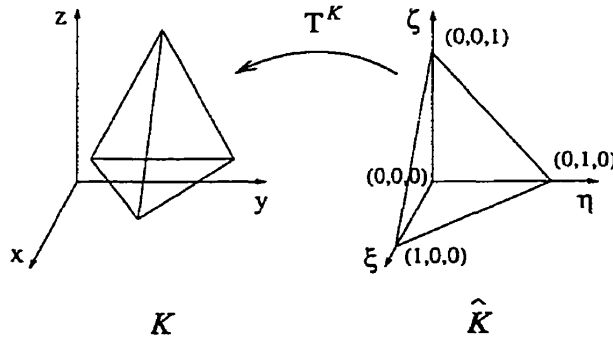


Figure 2.3: Passage d'un élément K à l'élément de référence \hat{K} en 3D

Ainsi on effectue toutes les intégrales seulement sur l'élément référentiel ce qui permet entre autres de pouvoir utiliser les quadratures de Gauss pour intégrer numériquement chacun des termes de formulation. Voir par exemple Fortin et Garon (2000) pour les transformations T^K ainsi que les quadratures de Gauss.

Le choix des fonctions d'interpolation dépend du choix du type d'éléments. Un choix populaire est l'élément de Taylor-Hood aussi connu sous le nom de P2-P1 (voir les figures 2.4-2.5). Cet élément vérifie la condition de Brezzi, aussi connue sous le nom de condition *inf-sup* qui assure l'existence et l'unicité de la solution en vitesse et en pression du problème de Stokes (voir Brezzi et Fortin (1991)).

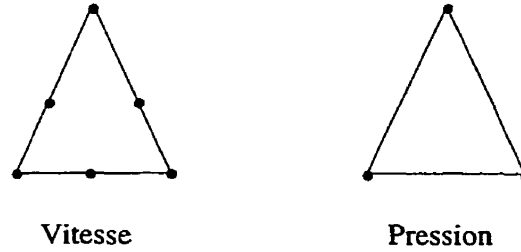


Figure 2.4: Éléments triangulaires P2-P1

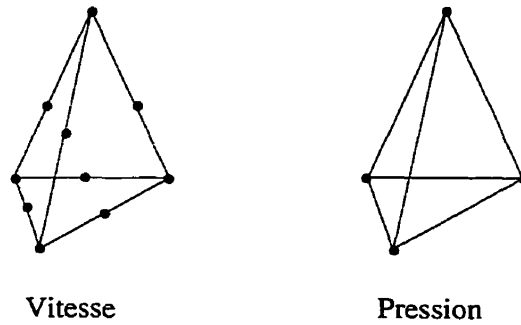


Figure 2.5: Éléments tétraédriques P2-P1

Avec cet élément P2-P1, on a donc $n_c^u = 6$ noeuds de calcul en vitesse et $n_c^p = 3$ noeuds de calcul en pression dans le cas bidimensionnel et $n_c^u = 10$ noeuds de calcul en vitesse et $n_c^p = 4$ noeuds de calcul en pression dans le cas tridimensionnel.

Comme son nom l'indique, avec l'élément P2-P1 on utilise des approximations quadratiques en vitesse et linéaires en pression. Par exemple, en deux dimensions, on prendrait les 6 fonctions quadratiques suivantes en vitesse:

$$\begin{aligned}
 \psi_{u,1}^K &= -(1 - \xi - \eta)(1 - 2(1 - \xi - \eta)) \\
 \psi_{u,2}^K &= -\xi(1 - 2\xi) \\
 \psi_{u,3}^K &= -\eta(1 - 2\eta)
 \end{aligned}$$

$$\begin{aligned}\psi_{u,4}^K &= 4\xi(1 - \xi - \eta) \\ \psi_{u,5}^K &= 4\xi\eta \\ \psi_{u,6}^K &= 4\eta(1 - \xi - \eta)\end{aligned}$$

et les 3 fonctions linéaires suivantes en pression:

$$\begin{aligned}\psi_{p,1}^K &= 1 - \xi - \eta \\ \psi_{p,2}^K &= \xi \\ \psi_{p,3}^K &= \eta\end{aligned}$$

De façon similaire, on pourrait se donner dix fonctions d'interpolation en vitesse et quatre en pression pour l'élément P2-P1, dans le cas tridimensionnel.

Un autre type d'élément qui est très souvent utilisé en raison du peu de noeuds de calcul qu'il comporte est l'élément *Mini* présenté aux figures 2.6-2.7.

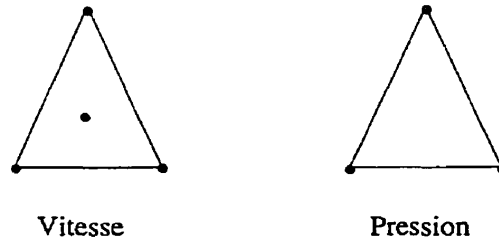


Figure 2.6: Éléments triangulaires *Mini*

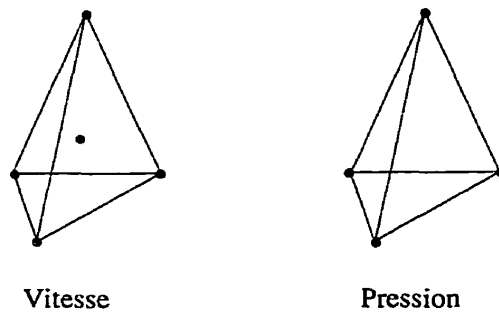


Figure 2.7: Éléments tétraédriques *Mini*

L'élément *Mini* est aussi appelé élément P1 bulle-P1 étant donné le noeud, en vitesse, au centre de l'élément qui amène la construction d'une fonction d'interpolation

qui s'annule sur tous les noeuds de l'élément sauf sur le noeud situé au barycentre. Cette fonction a donc un peu l'aspect d'une bulle.

Il existe plusieurs autres choix possibles d'éléments (voir Carey et Oden (1986)), mais nous nous contenterons de présenter les éléments P2-P1 et *Mini*. Notons que l'élément *Mini* est d'ordre 1 alors que celui de Taylor-Hood est d'ordre 2. Pour cette raison, nous avons décidé de ne travailler qu'avec ce dernier élément.

Par notre processus, nous avons donc construit autant de systèmes élémentaires qu'il y a d'éléments dans le maillage. Il faut maintenant regrouper toute l'information concernant ces systèmes dans une matrice globale, afin de résoudre ces systèmes globalement.

2.4 Assemblage de la matrice globale

Pour résoudre globalement les systèmes élémentaires, on assemble les contributions de chacun de ces systèmes dans une matrice globale. Il faut également fixer chacun des degrés de liberté en vitesse et en pression qui sont sur les parties de la frontière sur lesquelles on a imposé des conditions aux limites. Il y a des algorithmes simples qui permettent d'assembler la matrice globale en tenant compte de la connectivité entre chaque élément (voir Fortin et Geron (2000)). On est donc amené à résoudre le système global suivant:

$$\begin{bmatrix} (\hat{A} + \tilde{A}) & (B)^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta U \\ \delta P \end{bmatrix} = \begin{bmatrix} -\tilde{A} U_0 - B^T P_0 \\ -B U_0 \end{bmatrix} + \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (2.5)$$

En effet, lors de l'assemblage, toutes les contributions provenant des termes du vecteur S^K s'annulent deux à deux. Puisqu'entre deux éléments adjacents les vecteurs normaux sont opposés, ces contributions à la frontière des éléments sont donc les mêmes avec des signes opposés et s'annulent identiquement les uns avec les autres.

Notons que la matrice $\hat{A} + \tilde{A}$ est de dimension $n_D^u \times n_D^u$ alors que la matrice B est de dimension $n_D^p \times n_D^u$, n_D^u et n_D^p représentant le nombre total de degrés de liberté respectivement en vitesse et en pression sur tout le maillage.

On écrira le système 2.5 sous forme plus simple comme:

$$\begin{bmatrix} A_0 & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta U \\ \delta P \end{bmatrix} = \begin{bmatrix} R_1(U_0, P_0) \\ R_2(U_0, P_0) \end{bmatrix} \quad (2.6)$$

en introduisant les résidus vectoriels:

$$\begin{aligned} R_1(U_0, P_0) &= -\tilde{A}U_0 - B^T P_0 + F \\ R_2(U_0, P_0) &= -BU_0 \end{aligned}$$

et en posant:

$$A_0 = \hat{A} + \tilde{A}$$

Notons que nous avons ajouté un indice 0 à A_0 pour indiquer que A dépend de U_0 , ce qui sera important lors de la résolution du système par des méthodes itératives. En effet, à chaque itération, il faudra reconstruire la matrice puisque celle-ci dépend de la vitesse à l'itération précédente.

On verra au chapitre 3 comment on peut tirer profit de la structure de la matrice du système global 2.6, en découplant le calcul de δU de celui de δP .

Chapitre 3

Résolution du système global

3.1 Méthode d'Uzawa

La méthode d'Uzawa consiste à découpler la résolution du système 2.6 en deux étapes: en vitesse puis en pression.

Soit le système 2.6, réécrit sous la forme plus compacte suivante:

$$\begin{cases} A_0 \delta U + B^T \delta P = R_{1,0} & (3.1) \\ B \delta U = R_{2,0} & (3.2) \end{cases}$$

en notant les résidus $R_1(U_0, P_0)$ et $R_2(U_0, P_0)$ plus simplement comme $R_{1,0}$ et $R_{2,0}$.

Dans un premier temps, on isole δU de l'équation (3.1). On trouve:

$$\delta U = A_0^{-1}[R_{1,0} - B^T \delta P]$$

On l'introduit ensuite dans l'équation (3.2). Le système d'équations (3.1)-(3.2) devient alors équivalent à résoudre le système:

$$BA_0^{-1}B^T \delta P = BA_0^{-1}R_{1,0} - R_{2,0}$$

$$\text{avec } \delta U = A_0^{-1}[R_{1,0} - B^T \delta P]$$

Afin de simplifier la notation, nous allons poser:

$$\begin{aligned}\mathcal{A}_0 &= BA_0^{-1}B^T \\ G_0 &= BA_0^{-1}R_{1,0} - R_{2,0}\end{aligned}$$

On est donc amené à résoudre le système:

$$\mathcal{A}_0 \delta P = G_0 \quad (3.3)$$

$$\text{avec } \delta U = A_0^{-1}[R_{1,0} - B^T \delta P]$$

En pratique, il est très coûteux de calculer l'inverse d'une matrice. Il faut donc éviter à tout prix ce genre d'opération. Or dans les membres \mathcal{A}_0 et G_0 , on retrouve la matrice A_0^{-1} . Il est cependant possible de contourner ces deux calculs problématiques.

D'une part, on remarque que le membre G_0 s'écrit comme suit:

$$\begin{aligned}G_0 &= BA_0^{-1}R_{1,0} - R_{2,0} \\ &= BA_0^{-1}R_{1,0} + BU_0 \\ &= B[A_0^{-1}R_{1,0} + U_0]\end{aligned}$$

Pour éviter de calculer le terme $A_0^{-1}R_{1,0}$, on effectue plutôt l'opération équivalente qui consiste à résoudre le système $A_0 W = R_{1,0}$. C'est donc dire que:

$$G_0 = B[W + U_0]$$

où W est solution de $A_0 W = R_{1,0}$

Il est intéressant de remarquer que $W + U_0$ n'est rien d'autre qu'une approximation de la vitesse que nous noterons U_1^* , approximation qu'on obtient à l'aide de la pression initiale P_0 . Éventuellement, on pourra refaire le même calcul avec la pression corrigée P_1 , afin d'obtenir la vitesse corrigée.

En effet:

$$\begin{aligned}
 W + U_0 &= A_0^{-1} R_{1,0} + U_0 \\
 &= A_0^{-1} (-\tilde{A}U_0 - B^T P_0 + F) + U_0 \\
 &= A_0^{-1} (-\tilde{A}U_0 - \hat{A}U_0 + \hat{A}U_0 - B^T P_0 + F) + U_0 \\
 &= A_0^{-1} (-A_0 U_0 + \hat{A}U_0 - B^T P_0 + F) + U_0 \\
 &= -A_0^{-1} A_0 U_0 + A_0^{-1} \hat{A}U_0 - A_0^{-1} B^T P_0 + A_0^{-1} F + U_0 \\
 &= -U_0 + A_0^{-1} \hat{A}U_0 - A_0^{-1} B^T P_0 + A_0^{-1} F + U_0 \\
 &= A_0^{-1} (\hat{A}U_0 - B^T P_0 + F)
 \end{aligned}$$

ce qui est bien une approximation de U_1 obtenue à l'aide de la pression initiale, puisque le système 2.6 peut aussi s'écrire:

$$\begin{cases} (\hat{A} + \tilde{A}) (U_1 - U_0) + B^T (P_1 - P_0) = -\tilde{A}U_0 - B^T P_0 + F \\ B (U_1 - U_0) = -BU_0 \end{cases}$$

puisque $\delta U = U_1 - U_0$ et $\delta P = P_1 - P_0$.

On a ainsi:

$$\begin{cases} \hat{A}U_1 - \hat{A}U_0 + \tilde{A}U_1 - \tilde{A}U_0 + B^T P_1 - B^T P_0 = -\tilde{A}U_0 - B^T P_0 + F \\ BU_1 - BU_0 = -BU_0 \end{cases}$$

ou encore,

$$\begin{cases} \hat{A}U_1 + \tilde{A}U_1 + B^T P_1 = F + \hat{A}U_0 \\ BU_1 = 0 \end{cases}$$

donc,

$$\begin{cases} A_0 U_1 + B^T P_1 = F + \hat{A}U_0 & (3.4) \\ BU_1 = 0 & (3.5) \end{cases}$$

finalement, en isolant U_1 de (3.4) et en l'insérant dans (3.5), le système 2.6 est équivalent à résoudre:

$$B A_0^{-1} B^T P_1 = B A_0^{-1} (F + \hat{A}U_0)$$

$$\text{avec } U_1 = A_0^{-1} [F + \hat{A}U_0 - B^T P_1]$$

Tout ceci pour dire que le calcul de G_0 revient simplement au calcul suivant:

$$G_0 = B[W + U_0] = BU_1^* \equiv -R_{2,1}^*$$

c'est-à-dire que G_0 s'écrit comme $-R_2(U_1^*, P_1)$.

Notons qu'après avoir obtenu une nouvelle pression P_1 (en résolvant $\mathcal{A}_0 \delta P = G_0$), il est nécessaire de résoudre le système $\mathcal{A}_0 W = R_{1,1}$ avec le bon résidu $R_{1,1}$, obtenu à l'aide de P_1 , pour obtenir une vitesse U_1 .

D'autre part, il reste encore un problème causé par le calcul de la matrice \mathcal{A}_0^{-1} qu'on retrouve dans la matrice \mathcal{A}_0 . Notons tout d'abord que la matrice \mathcal{A}_0 est symétrique et définie positive, ce qui nous assure que la matrice $\mathcal{A}_0 = BA_0^{-1}B^T$ est aussi symétrique et définie positive (voir l'annexe A), puisque lorsqu'on utilise une discrétisation vérifiant la condition de Brezzi, la matrice B^T possède des colonnes linéairement indépendantes. Il est donc possible d'utiliser la méthode du gradient conjugué (voir annexe B) pour résoudre le système (3.3). Notons que l'algorithme de la méthode nécessite l'utilisation de la matrice du système uniquement pour faire un produit de celle-ci avec différents vecteurs (voir l'algorithme en page 58).

On peut donc contourner efficacement le problème dû à la matrice \mathcal{A}_0 du système $\mathcal{A}_0 \delta P = G_0$ en utilisant la méthode du gradient conjugué et en faisant ce qu'on appelle du *matrix-free*. Le principe du *matrix-free* est simple, on ne connaît pas explicitement la matrice \mathcal{A}_0 , par contre on connaît son effet lors d'un produit de la matrice par un vecteur.

Il s'agit donc de définir le produit de la matrice \mathcal{A}_0 par un vecteur quelconque u . Pour effectuer le produit $\mathcal{A}_0 u$, c'est-à-dire le produit $BA_0^{-1}B^T u$, on procède donc en trois étapes comme suit:

$$\left\{ \begin{array}{l} \text{Calculer } B^T u \rightarrow \text{mettre dans le vecteur } v \\ \text{Résoudre } \mathcal{A}_0 w = v \rightarrow \text{on obtient } w \quad (w = A_0^{-1}B^T u) \\ \text{Calculer } Bw \rightarrow \text{on obtient } BA_0^{-1}B^T u \end{array} \right.$$

On n'a donc qu'à adapter l'algorithme du gradient conjugué en utilisant les trois étapes précédentes pour faire les produits matrice-vecteur que l'on retrouve dans l'algorithme. On évite ainsi le calcul de l'inverse de la matrice \mathcal{A}_0 .

On peut montrer que la rapidité de la convergence des méthodes itératives comme la méthode du gradient conjugué pour résoudre le système $Ax = b$ dépend du conditionnement $\kappa(A)$ de la matrice A . Plus le conditionnement de la matrice est petit, plus la méthode itérative convergera rapidement (voir Lascaux et Théodor (1987)). À ce sujet, il est souvent utile de préconditionner un tel système avant de le résoudre. Le principe consiste à résoudre le système:

$$M^{-1}Ax = M^{-1}b$$

qui possède la même solution que le système:

$$Ax = b$$

si la matrice M est inversible.

La matrice de préconditionnement est choisie de telle sorte que $\kappa(M^{-1}A) < \kappa(A)$ et ainsi que le conditionnement du système soit amélioré (en effet plus $\kappa(A)$ est près de 1, meilleur est le conditionnement de la matrice).

Un choix possible, selon Carey et Oden (1986) pour la matrice de préconditionnement du système $\mathcal{A}_0 \delta P = G_0$ est la matrice masse M définie par:

$$(M)_{ij} = \int_{\Omega} \psi_{p,i}^K \psi_{p,j}^K d\mathbf{x}$$

pour $i, j = 1, \dots, n_D^p$ (le nombre total de degré de liberté en pression), la matrice \mathcal{A}_0 étant de dimension $n_D^p \times n_D^p$.

Il est également possible de prendre seulement la diagonale de la matrice M pour préconditionner le système.

On obtient finalement l'algorithme d'Uzawa de la page suivante pour résoudre le système global 2.6. Notons que puisque la matrice A_k dépend de la vitesse U_k à l'itération k , il est nécessaire d'assembler cette matrice à chaque itération, c'est-à-dire à chaque fois qu'on modifie la solution en vitesse.

Algorithme d'Uzawa pour résoudre le système 2.6

0 Initialisations

- | | |
|-----------------------------------|----------------|
| 1 Choisir un critère d'arrêt | ϵ |
| et un nombre maximum d'itérations | N_{max} |
| 2 Se donner | U_0 et P_0 |
| 3 Assembler | M et B |

Pour $i = 0, 1, 2, \dots, N_{max}$

1 Calcul de G_i

- | | |
|--------------------------------|-----------------------------------|
| 1 Assembler | A_i et $R_{1,i}$ |
| 2 Résoudre (gradient conjugué) | $J^{-1} A_i W = J^{-1} R_{1,i}$ |
| 3 Mettre à jour | $U_{i+1}^* = W + U_i$ |
| 4 Calculer | $G_i = BU_{i+1}^* \equiv R_{2,i}$ |

2 Résoudre en pression

- | | |
|--------------------------|--|
| 1 Résoudre (matrix-free) | $M^{-1} \mathcal{A}_i \delta P = M^{-1} G_i$ |
| 2 Mettre à jour | $P_{i+1} = P_i + \delta P$ |

3 Résoudre en vitesse

- | | |
|--------------------------------|-----------------------------------|
| 1 Assembler | $R_{1,i+1}$ |
| 2 Résoudre (gradient conjugué) | $J^{-1} A_i W = J^{-1} R_{1,i+1}$ |
| 3 Mettre à jour | $U_{i+1} = U_i + W$ |

si $\|R_{2,i}\| = \|-BU_i\| < \epsilon$, arrêt
sinon retour à l'étape 1.1

Les systèmes $J^{-1} A_i W = J^{-1} R_{u,i}$ sont résolus à l'aide du préconditionnement de Jacobi, d'où le nom de la matrice J (voir Lascaux et Théodor (1987) pour plus d'informations concernant le préconditionnement de Jacobi), alors que les systèmes $M^{-1} \mathcal{A}_i \delta P = M^{-1} G_i$ sont préconditionnés par la matrice masse présentée à page précédente.

3.2 Autres approches de résolution

3.2.1 Pénalisation

Nous avons investi un certain temps pour tenter d'apporter des améliorations à l'algorithme d'Uzawa précédent. Nous avons tenté d'ajouter un terme de pénalisation $\tau B^T B$ dans la matrice globale du système 2.6 comme suit:

$$\begin{bmatrix} A_i + \tau B^T B & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta U \\ \delta P \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

Le coefficient de pénalisation τ étant choisi près de 1. La solution de ce système est la même que celle du système sans pénalisation puisque $B \delta U = 0$. Le but de cet ajout était d'améliorer le conditionnement de la matrice A_i . La matrice $A_i^* = B (A_i + \tau B^T B)^{-1} B^T$ qui apparaîtrait dans l'algorithme d'Uzawa pénalisé étant mieux conditionnée que la matrice $A_i = B A_i^{-1} B^T$.

Cependant, après plusieurs essais et vérifications, il semble que l'ajout d'une telle pénalisation n'a pas donné les résultats escomptés. Le choix optimal, en terme de rapidité de convergence, pour le coefficient τ étant 0, c'est-à-dire lorsqu'il n'y a pas de pénalisation.

3.2.2 Stabilisation

Si on utilise l'élément *Mini*, il est possible de stabiliser la matrice globale en éliminant les degrés de libertés associés à la fonction bulle dont nous avons parlé précédemment. C'est ce qu'on appelle faire une condensation de la fonction bulle. Cette condensation mène à l'ajout d'une matrice dans la matrice globale et on doit résoudre le système suivant:

$$\begin{bmatrix} A_i & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} \delta U \\ \delta P \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

où la matrice C provient de la condensation.

À ce sujet, on peut se référer à l'article de Fortin et al. (2000) ainsi qu'aux articles de Wathen et Silvester (1993, 1994) qui présentent une vaste discussion sur différents préconditionneurs qu'on peut utiliser avec une telle stabilisation.

3.3 Visualisation des résultats

Après avoir résolu le système 2.6 par l'algorithme d'Uzawa, on obtient une solution en chaque noeud de calcul du maillage. Plus précisément, on obtient les composantes (2 ou 3 selon le cas) de la vitesse et la composante de la pression sur chacun des noeuds. Il ne reste plus qu'à visualiser les résultats.

Pour ce faire, on peut utiliser des logiciels tel que *Vu*, un outil de visualisation très puissant développé par le *CERCA*, le *Centre de recherche en calcul appliqué*. Les résultats présentés au chapitre suivant ont d'ailleurs été obtenus à l'aide du logiciel *Vu*.

Chapitre 4

Résultats numériques

Il existe plusieurs logiciels commerciaux et non commerciaux programmés pour résoudre, à l'aide de la méthode des éléments finis, différents problèmes d'équations différentielles. Le *GIREF*, le *Groupe Interdisciplinaire de Recherche en Éléments Finis*, un groupe de recherche de l'Université Laval, travaille depuis quelques années à mettre au point le code **MEF++**, un code d'éléments finis capable de résoudre des problèmes provenant de divers domaines de la physique. Le code **MEF++**, son nom l'indique, est programmé en langage C++. Nous avons donc utilisé ce code pour résoudre le problème décrit dans les trois premiers chapitres de ce mémoire.

Dans ce chapitre, nous présentons quelques résultats obtenus à l'aide de **MEF++**. On pourra se référer à l'annexe C pour une description de chacune des étapes du code *calculStokesGenNenUzawa* qui a permis de résoudre, par la méthode d'Uzawa, le problème de Stokes pour les fluides newtoniens généralisés. Les résultats graphiques ont été obtenus à l'aide du logiciel *Vu* du *CERCA*. Concernant *Vu* nous référons le lecteur à Ozell et Pic (1998).

4.1 Validation des résultats

4.1.1 Écoulement de *Poiseuille* dans un tube

Dans un premier temps, il est important de valider les résultats que nous obtenons avec le code. Pour ce faire, on se donne une géométrie très simple sur laquelle on

connaît la solution analytique pour un écoulement donné. On comparera ensuite la solution obtenue numériquement avec la solution analytique.

On peut montrer (voir Agassant et al. (1996)) que, pour un écoulement dit de *Poiseuille* avec viscosité modélisée à l'aide de la *loi de puissance*, c'est-à-dire un écoulement d'un fluide pseudoplastique (la viscosité étant $\eta(|\dot{\gamma}(\mathbf{u})|) = \eta_0 |\dot{\gamma}(\mathbf{u})|^{N-1}$) dans un tube de rayon R et de longueur L sur lequel on impose une différence de pression ΔP entre l'entrée et la sortie du tube, la solution analytique est donnée par:

$$\mathbf{u}(r) = \frac{N}{N+1} \left\{ \frac{1}{2\eta_0} \frac{\Delta p}{L} \right\}^{\frac{1}{N}} R^{\left(\frac{1+N}{N}\right)} \left\{ 1 - \left(\frac{r}{R} \right)^{\left(\frac{1+N}{N}\right)} \right\} \quad (4.1)$$

où r correspond à la position en coordonnées polaires.

Prenons donc un tube de longueur $L = 20$ et de rayon $R = 1$, suivant l'axe des x . Donnons-nous un maillage de ce tube comme suit:



Figure 4.1: Maillage d'un tube avec 6000 éléments (tubex2)

Il est possible de montrer (voir Fortin (1995)) que les paramètres du modèle de la *loi de puissance* pour une solution de 2% de polyisobutylène dans du primol 335 sont $\eta_0 = 228,34$, $N = 0,38$. La viscosité s'écrit donc:

$$\eta(|\dot{\gamma}(\mathbf{u})|) = 228,34 |\dot{\gamma}(\mathbf{u})|^{0,38-1}.$$

On résoud donc le problème de Stokes sur ce tube avec $\eta_0 = 228,34$, $N = 0,38$, (avec $C = 0$, $\lambda = 1$ et $\mathbf{f} = \mathbf{0}$). On impose les composantes u_2 et u_3 de la vitesse à 0 en entrée et en sortie et la vitesse nulle sur la paroi latérale du tube, à l'aide des conditions aux limites de *Dirichlet*. On impose les conditions limites de type *Neumann* à l'entrée et à la sortie du tube de telle sorte que l'on ait $\Delta p = 10000$. Plus précisément, on pose $(-pI + 2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u})) \cdot \mathbf{n} = (10000, 0, 0)$ en entrée (le vecteur normal est donc $\mathbf{n} = (-1, 0, 0)$), ce qui implique qu'on impose une pression de 10000 en entrée, puisque qu'on a imposé $u_2 = u_3 = 0$. En sortie, on impose $(-pI + 2\eta(|\dot{\gamma}(\mathbf{u})|) \dot{\gamma}(\mathbf{u})) \cdot \mathbf{n} = (0, 0, 0)$, (en sortie $\mathbf{n} = (1, 0, 0)$), c'est-à-dire une pression nulle.

On obtient la solution décrite à la figure 4.2. On peut observer le profil paraboloidal de la vitesse en différents endroits dans le tube. La pression, quant à elle, obéit à la pression qu'on a imposé, soit de 10000 en entrée et 0 en sortie. En effet, on obtient des pressions respectives de 10007.2 et -1.2373 .

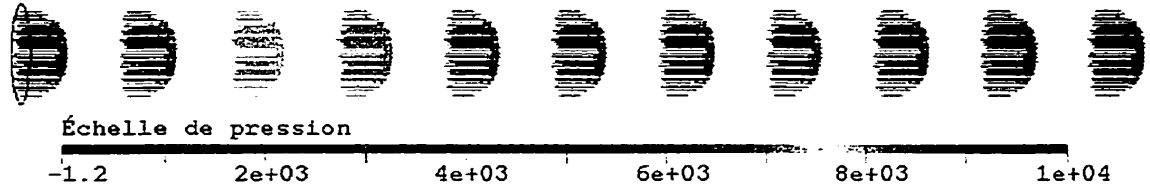


Figure 4.2: Solution numérique d'un écoulement de *Poiseuille* sur le tube

Afin de mieux apprécier le profil des vecteurs vitesse, on voit sur la figure 4.3 une coupe de la solution dans le plan $z = 0$.



Figure 4.3: Solution dans le plan $z = 0$ de l'écoulement de *Poiseuille* sur le tube

Comparons le profil de la solution analytique donnée à l'équation 4.1 avec celui de la solution numérique de la figure 4.2, obtenue sur le maillage composé de 6000 éléments. Étant donné les paramètres que nous avons choisis, la vitesse analytique suivant l'axe des x s'écrit, selon l'équation 4.1:

$$u(r) = 0,349 \{1 - r^{3,63}\}$$

où, dans ce cas, $r = \sqrt{y^2 + z^2}$.

On peut voir sur la figure 4.4 le profil de la solution analytique (en trait plein) ainsi que celui de la solution numérique (avec les o) sur le maillage avec 6000 éléments. Nous avons également effectué les calculs sur deux autres maillages; un maillage moins raffiné comprenant 750 éléments (avec les *) et un autre plus raffiné de 48000 éléments (avec les +). Pour raffiner le maillage, on a découpé en deux chacune des arêtes de chacun des éléments (il y a donc huit fois plus de tétraèdres sur un maillage raffiné). Le tableau suivant résume les informations concernant les trois maillages.

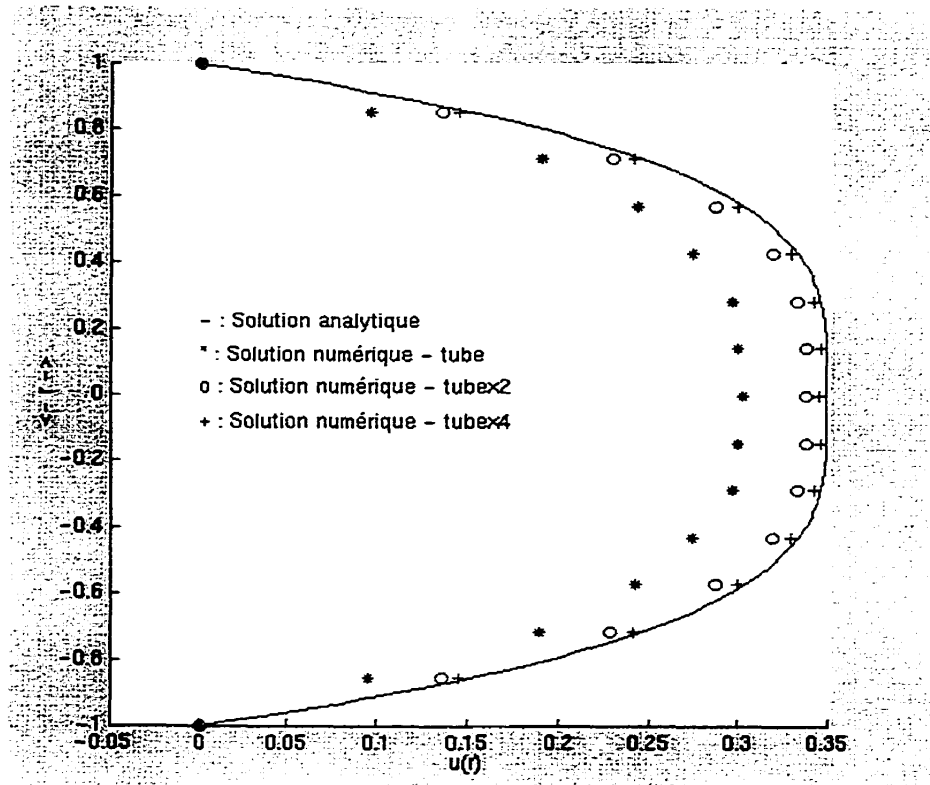


Figure 4.4: Solutions numériques et solution analytique d'un écoulement de *Poiseuille*

Il est intéressant de remarquer que plus le maillage est raffiné, plus la solution se rapproche de la solution analytique.

Tableau 4.1: Informations à propos des maillages tube, tubex2 et tubex4

Maillages tube, tubex2 et tubex4					
Maillage	Nombre d'éléments	Nombre de sommet	Nombre d'arêtes	n_D^u	n_D^p
tube	750	186	1035	3663	186
tubex2	6000	1221	7620	26523	1221
tubex4	48000	8841	58440	201843	8841
n_D^u, n_D^p : Nombre de degrés de liberté en vitesse et pression					

Il est important de remarquer que, même avec le maillage composé de 48000 éléments, on obtient une légère différence entre la solution analytique et les solutions obtenues par éléments finis. Cette erreur est due principalement à deux causes.

Premièrement, étant donné que nous utilisons des tétraèdres pour faire le maillage, le tube n'est en fait qu'une approximation d'un tube. On compare donc la solution d'un écoulement sur un véritable tube avec un écoulement sur un tube polygonal. C'est d'ailleurs pourquoi en raffinant le maillage du tube, la solution est améliorée. L'autre source d'erreur provient du fait qu'on utilise des interpolants quadratiques en vitesse (nous utilisons l'élément P2-P1), ce qui nous permet d'obtenir exactement des solutions de type quadratiques (et de degrés moins élevés) en vitesse. Cependant, dans notre cas, la vitesse analytique comporte le terme $r^{3,6337}$. En fait, il est impossible de capter la bonne solution pour des valeurs de N inférieures à 1 puisque dans ce cas l'exposant de r devient supérieur à 2. C'est donc dire qu'on ne peut obtenir la solution exacte à l'écoulement de *Poiseuille* dans le cas des fluides non newtoniens avec l'élément P2-P1.

4.1.2 Écoulement dans une cavité cubique

Un autre problème très classique est celui de la cavité cubique. On prend le domaine cubique $[0, 1]^3$ comme à la figure 4.5. On prend les paramètres $\eta_0 = 1, 0$, $N = 1, 0$ (donc C et λ quelconques) et on choisit $\mathbf{f} = \mathbf{0}$. On impose une vitesse nulle sur toute la frontière sauf sur la paroi supérieure (dans le plan $z = 1$) où on pose $\mathbf{u} = (1, 0, 0)$ (à l'aide des conditions aux limites de *Dirichlet*), ce qui aura pour effet d'induire une circulation du fluide dans la cavité.

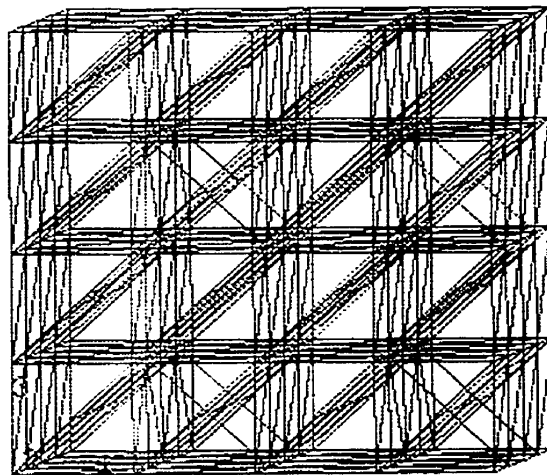


Figure 4.5: Maillage d'une cavité avec 384 éléments

Tableau 4.2: Informations à propos des maillages cube, cubex2 et cubex4

Maillages cube, cubex2 et cubex4					
Maillage	Nombre d'éléments	Nombre de sommet	Nombre d'arêtes	n_D^u	n_D^p
cube	384	125	604	2187	125
cubex2	3072	729	4184	14739	729
cubex4	24576	4913	31024	107811	4913
n_D^u, n_D^p : Nombre de degrés de liberté en vitesse et pression					

On obtient la solution présentée à la figure 4.7 qui à été obtenue à l'aide du maillage cubex2. On observe une zone de recirculation à l'intérieur de la cavité.

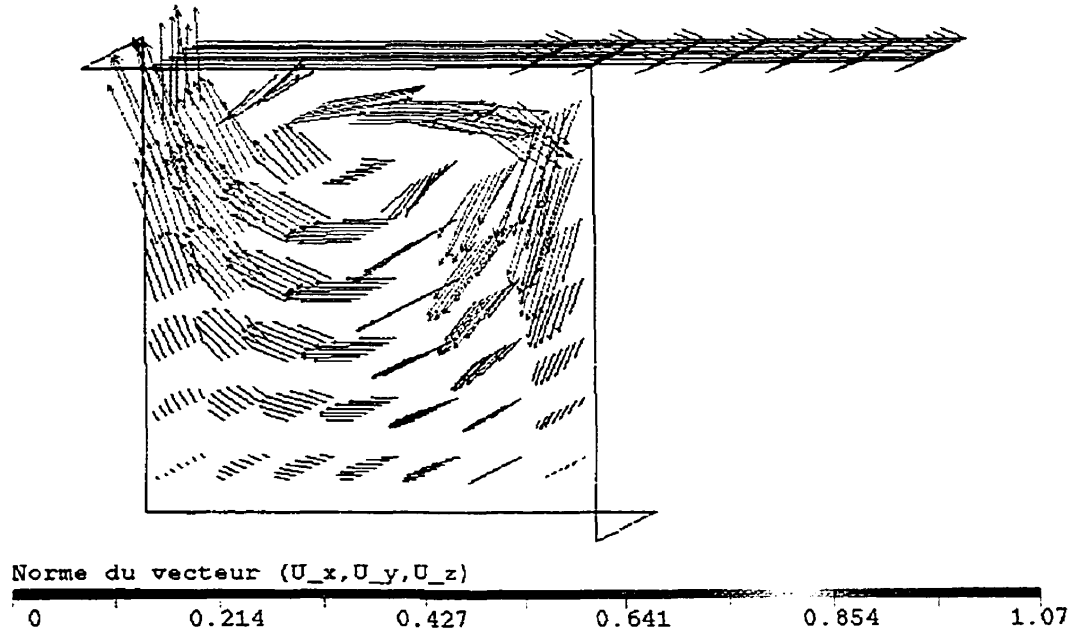


Figure 4.6: Solution numérique d'un écoulement induit dans la cavité (solution obtenue avec 3072 éléments)

Nous avons également effectué les calculs sur un maillage plus raffiné, la maillage cubex4. Afin de pouvoir mieux visualiser la solution, on présente à la figure 4.7 seulement une coupe de la solution dans le plan $y = \frac{1}{2}$.

On peut voir sur la figure 4.8 les isovaleurs de la composante en x de la vitesse dans le plan $y = \frac{1}{2}$.

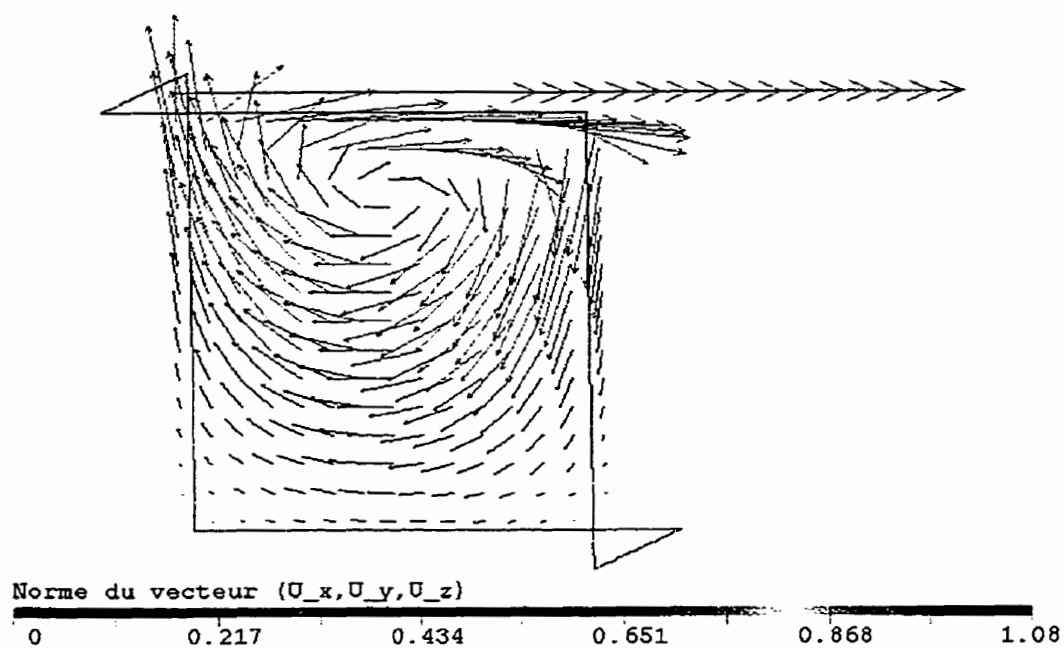


Figure 4.7: Solution numérique d'un écoulement induit dans la cavité (solution obtenue avec 24576 éléments)

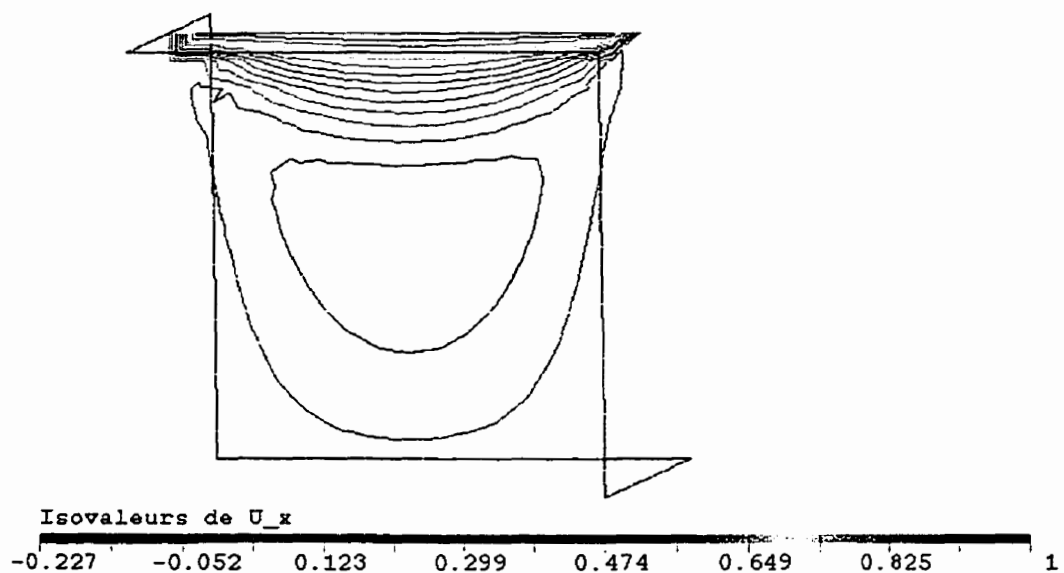


Figure 4.8: Isovaleurs de la composante x de la vitesse, dans la cavité

Nous allons utiliser la cavité afin de montrer le comportement de la méthode du gradient conjugué lors de la résolution du système $\mathcal{A}_0 \delta P = G_0$ (étape 2.1 de l'algorithme d'Uzawa). Étant donné que nous avons choisi $N = 1$, le problème à

résoudre est linéaire et une seule itération de la méthode de Newton est suffisante pour atteindre la convergence (nous avons choisi $\epsilon = 1 \times 10^{-6}$). Nous présentons sur le graphique de la figure 4.9, la valeur de la norme infinie du résidu préconditionné, $\|R_{2,i}\|_\infty$, en fonction des itérations de la méthode du gradient conjugué. Il a fallu

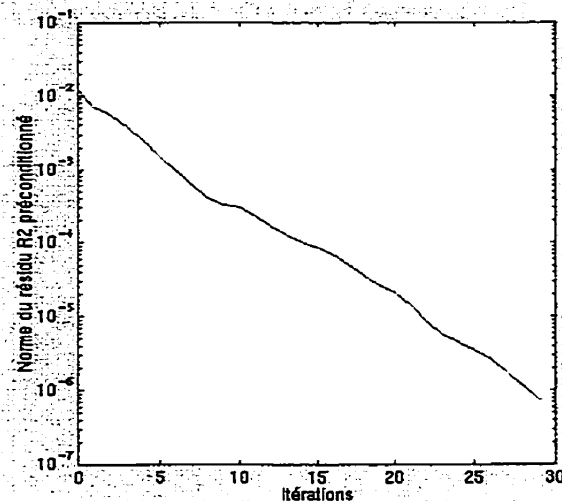


Figure 4.9: Norme infinie du résidu en fonction des itérations dans le cas linéaire

environ 30 itérations pour atteindre la convergence et ce nombre d'itérations est indépendant du raffinement du maillage.

Terminons cette section en parlant de la stratégie qui nous a permis de bien résoudre le problème de Stokes. La valeur de l'indice de pseudoplasticité N joue un rôle important quant à la complexité du problème qu'il faut résoudre. Quand $N = 1$, le problème est linéaire et il est relativement simple à résoudre. Par contre, plus N est petit plus le problème est difficile à résoudre, en raison de la non-linéarité. Par exemple, pour résoudre l'écoulement de *Poiseuille* sur le tube, avec $N = 0,38$, nous n'avons pas lancé les calculs directement avec $N = 0,38$ car la convergence est difficile à atteindre. Nous avons plutôt résolu le problème avec $N = 1$, pour ensuite abaisser la valeur de N graduellement, par exemple en prenant successivement $N = 0,8$; $0,6$; $0,45$ et $0,38$, en prenant comme solutions initiales, pour chacun des calculs, la solution obtenue avec le N précédent. Cette stratégie semble être la meilleure solution pour traiter le problème de non-linéarité présent dans la résolution du problème de Stokes pour les fluides newtoniens généralisés avec un indice de pseudoplasticité assez petit.

4.2 Autre géométrie

Dans cette section, nous allons considérer des écoulements de fluides sur une géométrie un peu plus complexe. En effet, il est important de s'assurer que notre code est capable de résoudre les équations de Stokes sur des géométries moins simples qu'un tube ou une cavité cubique. Parfois un code d'éléments finis rend de bons résultats sur des géométries simples mais a des difficultés à le faire sur des géométries complexes.

Nous allons maintenant considérer une contraction, une géométrie qui permettra de tester le code utilisé face à une géométrie un peu plus complexe. Donnons-nous le maillage d'une contraction tel qu'illustré à la figure suivante.

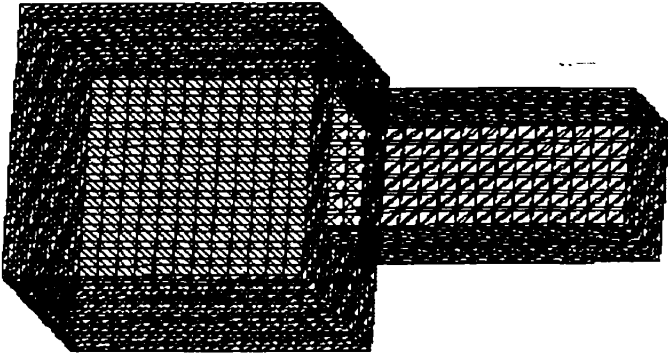


Figure 4.10: Maillage d'une contraction avec 30720 éléments

Tableau 4.3: Informations à propos du maillage Contraction

Maillage Contraction					
Maillage	Nombre d'éléments	Nombre de sommet	Nombre d'arêtes	n_D^u	n_D^p
Contraction	30720	6209	38976	135555	6209
n_D^u, n_D^p : Nombre de degrés de liberté en vitesse et pression					

Nous allons tenter de modéliser à nouveau l'écoulement d'une solution de 2% de polyisobutylène dans du primol 335, mais cette fois à l'aide de la loi de *Carreau*. On peut montrer (voir Fortin (1995)) que les paramètres du modèle de *Carreau* pour ce fluide pseudoplastique sont $\eta_0 = 4926,08$, $N = 0,331$, $\lambda = 116,922$ ($C = 1$). La

viscosité s'écrit donc:

$$\eta(|\dot{\gamma}(\mathbf{u})|) = 4926,08(1 + 116,922^2|\dot{\gamma}(\mathbf{u})|^2)^{\frac{0,331-1}{2}}$$

Réolvons donc les équations de Stokes sur cette contraction, avec les paramètres ci-hauts pour modéliser la viscosité du fluide considéré. Nous prendrons encore $\mathbf{f} = \mathbf{0}$. Nous imposons la vitesse nulle sur chacune des faces sauf à l'entrée et à la sortie ou on impose plutôt respectivement des pressions de 10000 et de 0. On obtient le solution décrite à la figure 4.11 (dans le plan $x = 0$).

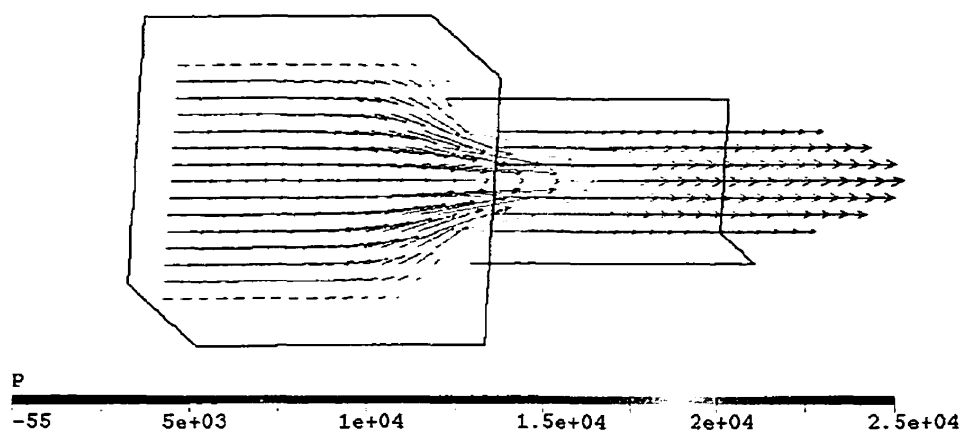


Figure 4.11: Solution numérique d'un écoulement dans la contraction

La figure 4.12 présente les isovaleurs de la pression.

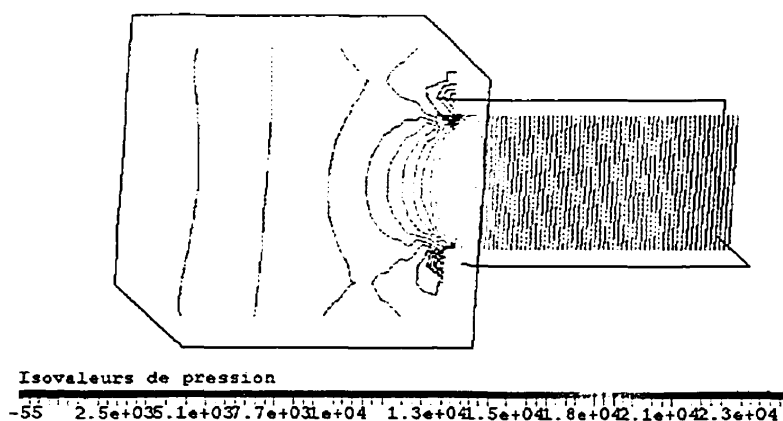


Figure 4.12: Isovaleurs de la pression, dans la contraction

4.3 Problèmes rencontrés en industrie

Voyons maintenant un problème plus près de ceux qu'on rencontre en industrie. À tous les jours, on utilise un tas d'objets fabriqués à partir de matières plastiques, par mise en forme de polymères. Pour produire la plupart de ces objets, on utilise différents procédés tels le calandrage, le soufflage, l'extrusion et l'injection.

Selon Agassant et al. (1996), l'extrusion est l'un des procédés les plus importants. Son principe consiste à transformer, à l'aide d'une vis à extrusion, de la matière plastique solide en un polymère fondu qui pourra ensuite être moulé. La vis à extrusion telle qu'illustrée à la figure 4.13, qui a été tirée du livre d'Agassant et al. (1996), permet donc de faire fondre, de mélanger et de donner une pression au polymère fondu permettant à celui-ci d'être injecté sous pression dans un moule, en un produit final.

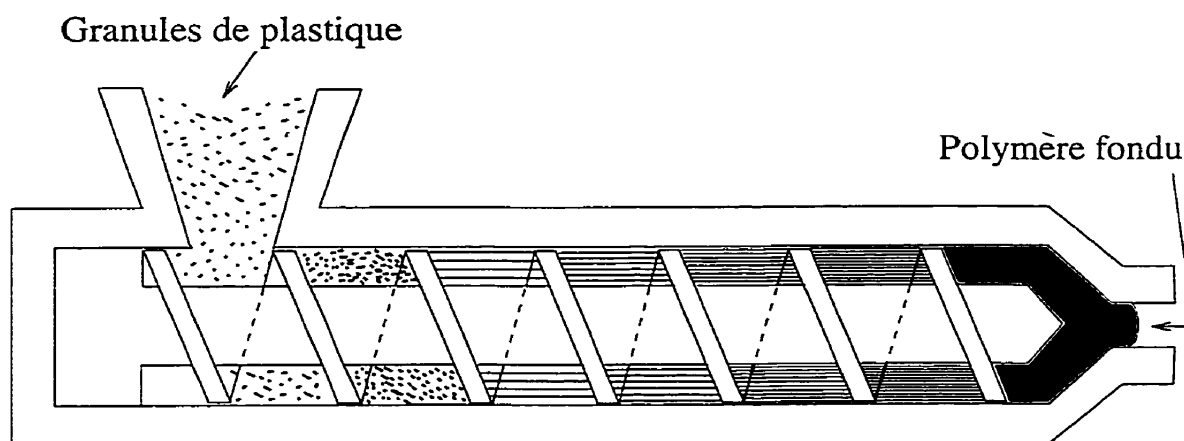


Figure 4.13: Schéma d'une extrudeuse avec une monovis

Encore une fois, nous allons référer le lecteur à Agassant et al. (1996) pour plus de détails sur les procédés d'extrusion et autres procédés de mise en forme de matières plastiques. L'article de Bravo et al. (1998) présente des simulations d'écoulement dans différents types de vis à extrusion, principalement des bi-vis à extrusion. Il est à noter qu'ils ont utilisé des méthodes directes pour résoudre les systèmes d'équations provenant de la discrétisation.

Nous allons donc tenter d'étudier l'écoulement d'un fluide pseudoplastique à l'intérieur d'une vis à extrusion, avec notre méthode.

Mentionnons immédiatement que nous ne prétendons pas faire une vraie simulation d'un écoulement dans un vis d'extrusion. D'une part, nous avons considéré que la vis était fixe, alors qu'il faudrait imposer une vitesse de rotation à la vis (il faudrait tout simplement imposer des vitesses correspondantes sur les parois intérieures de la vis, à l'aide des conditions aux limites). D'autre part, comme nous le verrons, nous n'avons pas choisi un fluide qu'il serait réaliste de retrouver dans un procédé d'extrusion. Nous allons tout de même modéliser l'écoulement d'un fluide à l'intérieur d'une telle vis (sans rotation) de manière à s'assurer que notre méthode fonctionne bien sur un maillage d'assez grande complexité.

Pour ce faire, on maille l'intérieur d'une section de la vis comme on peut le voir aux figures 4.14 et 4.15. Il faut bien comprendre que le maillage est fait entre les dents de la vis, là où le fluide circule. Il faut aussi imaginer que cette vis est enfermée dans un fourreau (le tube qui entoure la vis).

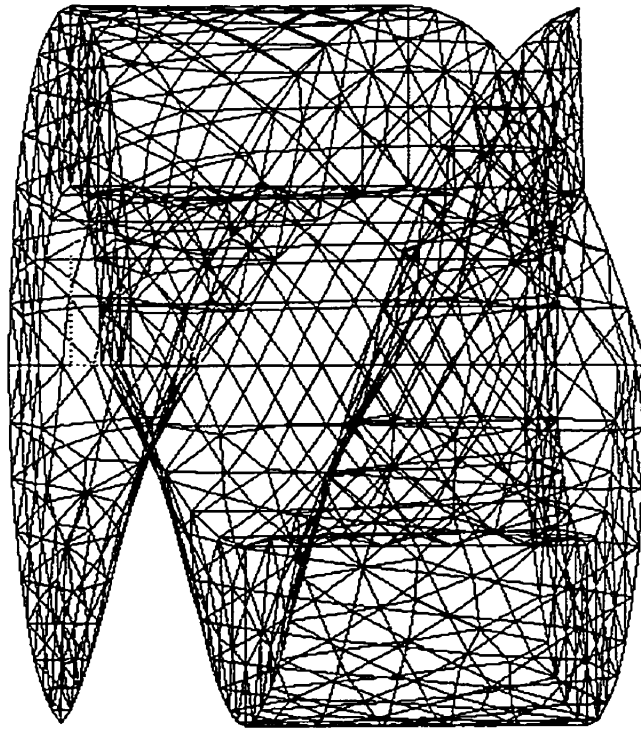


Figure 4.14: Maillage entre les dents d'une vis à extrusion - Vue de face

Considérons, cette fois-ci, une solution d'oxyde polyéthylène dans une autre solution de 50% d'eau et 50% de glycérine. Selon Carreau et al. (1997) les paramètres pour la loi de *Carreau* d'une telle solution sont: $\eta_0 = 102,0$, $N = 0,375$, $\lambda = 4,36$.

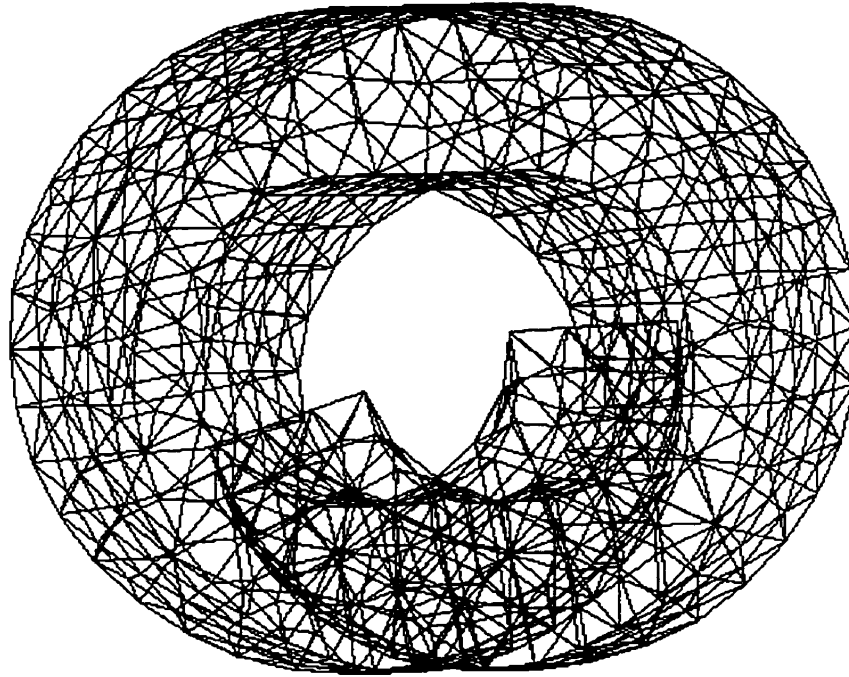


Figure 4.15: Maillage entre les dents d'une vis à extrusion - Vue de biais

Tableau 4.4: Informations à propos du maillage Vis

Maillage Vis					
Maillage	Nombre d'éléments	Nombre de sommet	Nombre d'arêtes	n_D^u	n_D^p
Vis	3769	1008	5419	19281	1008
n_D^u, n_D^p : Nombre de degrés de liberté en vitesse et pression					

Nous allons donc résoudre les équations de Stokes avec ces paramètres et $\mathbf{f} = \mathbf{0}$ en imposant une vitesse nulle sur la vis et sur le fourreau. Pour qu'il y ait un écoulement, on impose une différence de de pression $\Delta P = 100$ entre la partie avant et la partie arrière de la vis.

On obtient la solution décrite aux figures 4.16 et 4.17 des pages suivantes. Nous avons donné les vecteurs vitesses dans dans différents plans. Sur la figure 4.18, on peut observer la solution sur toute la vis.

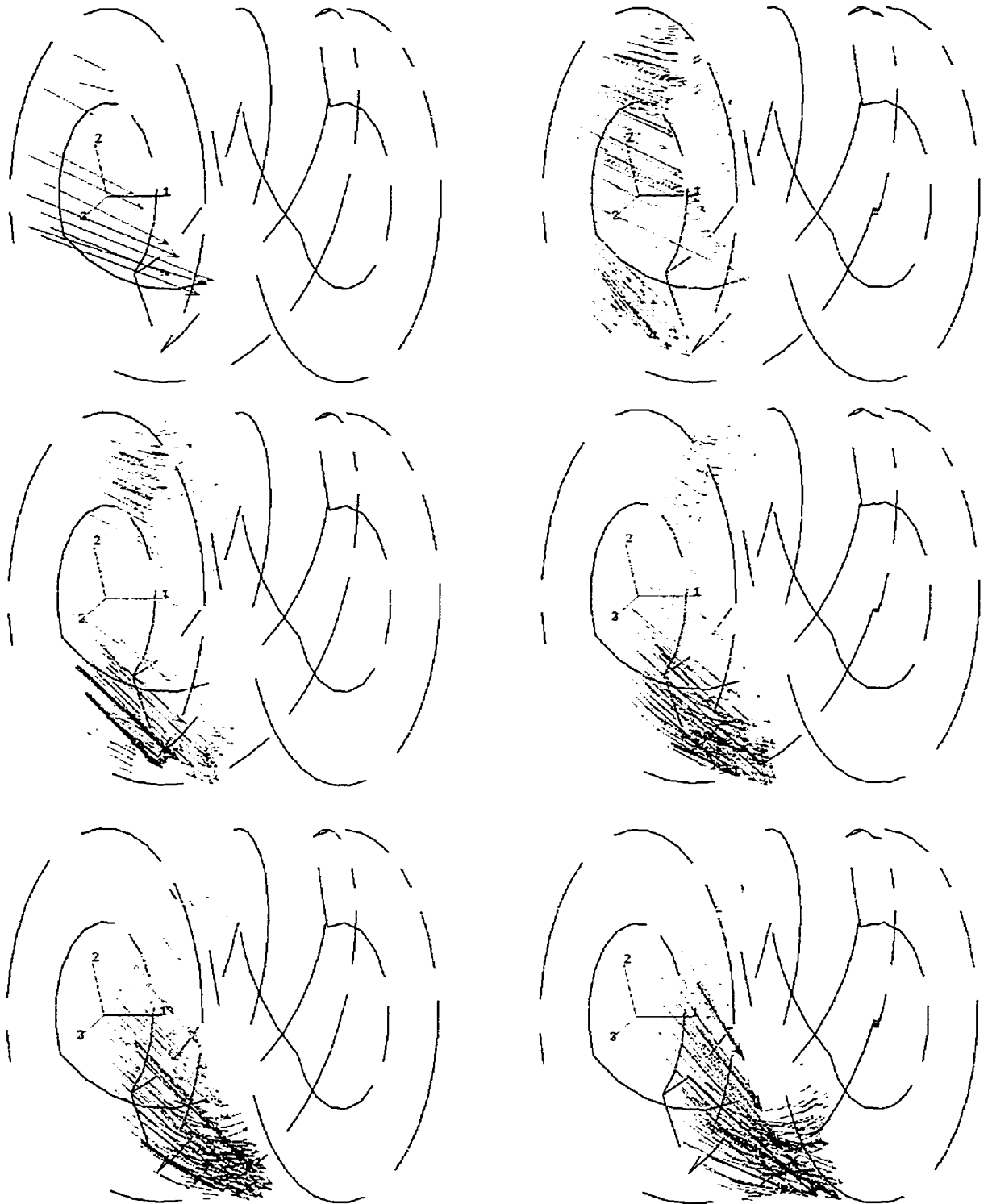


Figure 4.16: Solution de l'écoulement dans la vis, dans différents plans



Figure 4.17: Solution de l'écoulement dans la vis, dans différents plans (suite)



Figure 4.18: Solution de l'écoulement dans la vis

Bien que nous n'ayons pas la solution analytique d'un tel écoulement dans une vis, on peut quand même affirmer que la solution semble visuellement correcte. En effet, on peut aisément imaginer qu'un fluide puisse avoir un comportement comme celui décrit aux figures précédentes.

Suite aux résultats que nous avons présentés dans ce chapitre, nous pouvons conclure que la méthode des éléments finis permet de bien résoudre le problème de Stokes. Nous avons su bien traiter la non-linéarité découlant de l'utilisation des modèles *Loi de puissance* et de *Carreau* pour décrire la viscosité. Finalement la méthode d'Uzawa s'est avérée une méthode efficace pour la résolution des systèmes d'équations.

Conclusion

Dans ce mémoire, nous avons tenté de simuler des écoulements de fluides incompressibles newtoniens et non newtoniens. Pour ce faire, nous avons résolu les équations de Stokes en utilisant deux modèles (*loi de puissance* et *modèle de Carreau*) pour décrire le comportement de la viscosité des fluides. Nous avons présenté la méthode des éléments finis spécifiquement appliquée à la résolution des équations (non linéaires) de Stokes pour les fluides non newtoniens ainsi que la méthode d'Uzawa pour résoudre les systèmes d'équations provenant de cette dernière méthode.

Après avoir validé la méthode que nous avons utilisée, en résolvant le problème de Stokes sur des géométries simples, dont on connaît la solution analytique, nous avons résolu les équations de Stokes sur des géométries d'une plus grande complexité, afin de voir si le code était robuste face à des maillages complexes. Par la suite, nous avons essayé de résoudre un problème du type de ceux que l'on pourrait rencontrer en industrie. Nous avons fait une brève incursion du côté des procédés d'extrusion pour tenter d'appliquer notre méthode à de vrais problèmes. On a donc fait la simulation d'écoulements de polymères et d'autres fluides non newtoniens dans une vis à extrusion.

Il s'est donc avéré qu'il était possible grâce à la méthode des éléments finis combinée à la méthode d'Uzawa, de bien résoudre les équations de Stokes pour des fluides newtoniens généralisés, en utilisant les modèles *loi de puissance* et *modèle de Carreau* pour décrire la viscosité de ces fluides. Chose intéressante avec la méthode d'Uzawa, c'est qu'on est amené à résoudre un système d'équations linéaires dont la matrice est symétrique et définie positive. Ainsi, nous avons pu utiliser la méthode du gradient conjugué ce qui nous a permis de résoudre des systèmes de grandes tailles que l'on obtient lorsqu'on résout des problèmes tridimensionnels sur des maillages contenant

beaucoup d'éléments.

Comme nous en avons fait mention dans le chapitre 3, il semble qu'une pénalisation du type $A + rB^T B$ n'a pas donné les résultats voulus quant à la résolution, par la méthode d'Uzawa, des systèmes d'équations provenant de la discrétisation des équations de Stokes par la méthode des éléments finis. Nous pensions qu'une telle pénalisation de la matrice du système global améliorerait la convergence, mais on a dû constater que la convergence était meilleure sans cette pénalisation.

Suite aux travaux concernant ce mémoire, en ce qui concerne les perspectives d'avenir dans le domaine de la mécanique des fluides et pour donner suite aux résultats présentés dans ce mémoire, il faudrait simuler des écoulements plus réalistes sur les vis d'extrusion en rotation. Également, il serait possible de faire des simulation sur des doubles vis d'extrusion. Dans un autre ordre d'idées, il serait envisageable de tenter de résoudre les équations de Navier-Stokes pour les fluides newtoniens généralisés. Il y a également tout le concept de l'adaptation de maillage, que nous n'avons pas traité ici.

Références

- AGASSANT, J.-F., AVENAS, P., SERGENT, J.-P., VERGNES, B. et VINCENT, M. (1996). *La mise en forme des matières plastiques*. Techniques & Documentation Lavoisier, Paris.
- BERTRAND, F., TANGUY, P. A., BRITO DE LA FUENTE, E. et CARREAU, P. (1999). Numerical modeling of the mixing flow of second-order fluids with helical ribbon impellers. *Comput. Methods Appl. Mech. Engrg*, 180, 267–280.
- BRAVO, V. L., HRYMAK, A. N. et WRIGHT, J. D. (1998). Numerical simulation of pressure and velocity profiles in mixing elements of a co-rotating twin screw extruder. *Polymer Engineering & Science*, 25 pages. Article accepté en mars 1999.
- BREZZI, F. et FORTIN, M. (1991). *Mixed and Hybride Finite Element Methods*. Springer-Verlag, New-York.
- CAREY, G. F. et ODEN, J. T. (1986). *Finite Elements - Fluid Mechanics*, vol. VI. Prentice-Hall, Englewood Cliffs.
- CAREY, G. F., WANG, K. C. et JOUBERT, W. D. (1989). Performance of iterative methods for Newtonian and generalized Newtonian flows. *Int. J. Numer. Meth. Fluids*, 9, 127–150.
- CARREAU, P. J., DE KEE, D. et CHHABRA, R. P. (1997). *Rheology of Polymeric Systems: Principles and Applications*. Hanser, Cincinnati.
- CROCHET, M. J., DAVIES, A. R. et WALTERS, K. (1984). *Numerical Simulation of Non-Newtonian Flow*. Elsevier, Amsterdam.
- FORTIN, A. (1995). *Analyse numérique pour ingénieurs*. Éditions de l'École Polytechnique de Montréal, Montréal.

- FORTIN, A. et GARON, A. (2000). *Les éléments finis: de la théorie à la pratique*. À paraître.
- FORTIN, A., GUÉNETTE, R., LABBÉ, J. et MARCOTTE, J.-P. (2000). Iterative solvers for quadratic discretisation of the stokes problem. À paraître.
- GARTLING, D. K. (1986). Finite Element Methods for Non-Newtonian Flows. *Sandia Report SAND85-1704*.
- GREENBAUM, A. (1997). *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia.
- LASCAUX, P. et THÉODOR, R. (1987). *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, vol. 2. Masson, Paris.
- OZELL, B. et PIC, C. (1998). *Manuel d'utilisation du logiciel configurable de visualisation scientifique VU*. CERCA.
- PIRONNEAU, O. (1988). *Méthode des éléments finis pour les fluides*. Masson, Paris.
- REDDY, J. N. (1993). *An introduction to the Finite Element Method*. McGraw Hill, New York, seconde édition.
- ROBICHAUD, M. P., TANGUY, P. A. et FORTIN, M. (1990). An iterative implementation of the Uzawa algorithm for 3d fluidflow problems. *Int. J. Numer. Meth. Fluids*, 10, 429–442.
- TANGUY, P. A., FORTIN, A. et BERTRAND, F. (1988). A Three-dimensional Finite Element Model for Polymer Melt Flow. *Adv. Polym. Tech.*, 8, 99–113.
- WATHEN, A. J. et SILVESTER, D. J. (1993). Fast iterative solution of stabilised stokes systems. part i: Using simple diagonal preconditioners. *SIAM J. Numer. Anal.*, 30, pp. 630–649.
- WATHEN, A. J. et SILVESTER, D. J. (1994). Fast iterative solution of stabilised stokes systems. part ii: Using general block preconditioners. *SIAM J. Numer. Anal.*, 31, pp. 1352–1367.

Annexe A

Matrice symétrique et matrice définie positive

Dans ce qui suit $A = A_{n \times n}$ est une matrice de dimension n par n et $B = B_{m \times n}$ est une matrice de dimension m par n (avec $m \leq n$).

Définition 1 Une matrice A est dite symétrique si $A^T = A$.

Théorème 1 Si A est une matrice inversible et symétrique, alors la matrice A^{-1} est aussi une matrice symétrique.

Preuve On a:

$$\begin{aligned} (A^{-1})^T &= \left\{ \frac{1}{\det(A)} [\text{cof}(A)]^T \right\}^T \\ &= \frac{1}{\det(A)} \text{cof}(A) \\ &= \frac{1}{\det(A)} [\text{cof}(A)]^T \\ &= A^{-1} \end{aligned}$$

puisque la matrice des cofacteurs, $\text{cof}(A)$, est symétrique si A est symétrique. \square

Théorème 2 Si A est une matrice inversible et symétrique et B est une matrice quelconque, alors la matrice $B A^{-1} B^T$ est aussi une matrice symétrique.

Preuve On a:

$$\begin{aligned}(B A^{-1} B^T)^T &= (B^T)^T (A^{-1})^T (B)^T \\ &= B (A^{-1})^T B^T \\ &= B A^{-1} B^T\end{aligned}$$

puisque $(A^{-1})^T = A^{-1}$ lorsque A est symétrique. \square

Définition 2 Une matrice A est dite définie positive si

$$\langle A x, x \rangle > 0 \text{ pour tout } x \neq 0$$

avec $\langle A x, x \rangle = 0$ seulement si $x = 0$.

(\langle, \rangle signifie le produit scalaire $\langle x, y \rangle = x \cdot y^T$)

Théorème 3 Si A est une matrice définie positive (donc inversible), alors la matrice A^{-1} est aussi une matrice définie positive.

Preuve On a:

$$\begin{aligned}\langle A^{-1} x, x \rangle &= \langle y, A y \rangle \quad (\text{en posant } y = A^{-1} x) \\ &= \langle A y, y \rangle \\ &> 0\end{aligned}$$

puisque A est définie positive. De plus, $\langle A y, y \rangle$ sera égale à 0 seulement si $y = 0$, c'est-à-dire si $A^{-1} x = 0$, donc seulement si $x = 0$. \square

Théorème 4 Si A est une matrice inversible et définie positive et B est une matrice telle que B^T a des colonnes linéairement indépendantes, alors la matrice $B A^{-1} B^T$ est aussi une matrice définie positive.

Preuve On a:

$$\begin{aligned} \langle B A^{-1} B^T x, x \rangle &= \langle A^{-1} B^T x, B^T x \rangle \\ &= \langle A^{-1} z, z \rangle \quad (\text{en posant } z = B^T x) \\ &> 0 \end{aligned}$$

puisque A est définie positive, en utilisant le théorème 3. De plus, $\langle A^{-1} z, z \rangle$ sera égale à 0 seulement si $z = 0$, c'est-à-dire si $B^T x = 0$, donc seulement si $x = 0$, puisque les colonnes de B^T sont linéairement indépendantes. \square

Théorème 5 *Si A est une matrice symétrique et définie positive et B est une matrice telle que B^T a des colonnes linéairement indépendantes, alors la matrice $B A^{-1} B^T$ est aussi une matrice symétrique définie positive.*

Preuve On n'a qu'à utiliser les théorèmes 1 à 4. \square

Annexe B

Méthode du gradient conjugué

La méthode du gradient conjugué est une méthode itérative pour la résolution de systèmes d'équations linéaires tel le système $\mathcal{A}x = G$.

Principe de la méthode:

Il faut que la matrice \mathcal{A} soit symétrique et définie positive.

Trouver la solution du système $\mathcal{A}x = G$ est équivalent à trouver le point x qui minimise la fonctionnelle $J(x)$ définie par:

$$J(x) = \frac{1}{2} \langle \mathcal{A}x, x \rangle - \langle G, x \rangle$$

où \langle , \rangle signifie le produit scalaire $\langle x, y \rangle = x \cdot y^T$

Il s'agit de partir d'une solution initiale $x^{(0)}$ et de passer du vecteur $x^{(k)}$ au vecteur $x^{(k+1)}$ par la relation:

$$x^{(k+1)} = x^{(k)} + \mu^{(k)} u^{(k)}$$

$u^{(k)}$ étant une direction conjuguée de $u^{(1)}, u^{(2)}, \dots, u^{(k-1)}$ et $\mu^{(k)}$ étant choisi de façon à rendre $J(x^{(k)})$ minimum.

L'algorithme de la méthode du gradient conjugué est décrit à la page suivante.

Algorithme de la méthode du gradient conjugué:

• Initialisations:

On se donne $x^{(0)}$, un critère d'arrêt ϵ et un nombre maximum d'itérations N_{max} .

On pose $r^{(0)} = \mathcal{A}x^{(0)} - G$ et $u^{(0)} = -r^{(0)}$

• Itérations:

Pour $k = 0, 1, 2, \dots$

$$(1) \quad \mu^{(k)} = \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle u^{(k)}, \mathcal{A}u^{(k)} \rangle}$$

$$(2) \quad x^{(k+1)} = x^{(k)} + \mu^{(k)} u^{(k)}$$

$$(3) \quad r^{(k+1)} = \mathcal{A}x^{(k+1)} - G = r^{(k)} + \mu^{(k)} \mathcal{A}u^{(k)}$$

$$(4) \quad u^{(k+1)} = -r^{(k+1)} + \frac{\langle r^{(k+1)}, r^{(k+1)} \rangle}{\langle r^{(k)}, r^{(k)} \rangle} u^{(k)}$$

si $\langle r^{(k)}, r^{(k)} \rangle < \epsilon$ ou si $k = N_{max}$, arrêt

sinon retour à (1)

Pour plus de détails sur la méthode du gradient conjugué, on peut se référer à Lascaux et Théodor (1987)

Annexe C

Approche informatique

Dans cette annexe, nous nous proposons de faire un survol des différentes parties du code *calculStokesNewGenUzawa* programmé avec **MEF++** qui ont permis de résoudre le problème de Stokes pour les fluides newtoniens généralisés, à l'aide de la méthode d'Uzawa. Nous allons décrire les classes (notion de programmation orienté objet en C++) qui constitue le code *calculStokesNewGenUzawa* qui nous ont permis d'arriver à notre but.

Une des premières étapes que l'on retrouve dans le code est la déclaration des constantes pour les différents paramètres du modèle viscosité des fluides newtoniens généralisés: η_0 , C , λ et N et de leur assigner des valeurs (voir l'équation 1.2 de la page 9).

```
int main(int argc, char *argv[]) {
```

```
    DReel  lViscONewGen, lCNewGen, lLaNewGen, lNNewGen;
```

On se donne ensuite des objets de classe *Maillage*, classe qui contiendra toute l'information sur le maillage puis on lit les données concernant le maillage. Le paramètre *lNom* fait référence à un fichier contenant le maillage (exemple: *tubeR.mail*).

```
    Maillage lMail;
```

```
    lMsg = lMail.importe(lNom);
```

On déclare le type de frontière des éléments. Dans le cas linéaire, (*ChampGeoLin*) les éléments sont droits mais on pourrait aussi avoir des éléments avec des frontières constituées de courbes.

```
ChampGeoLin lChampGeo(lMail);
```

On se donne un objet de classe *Geometrie* qui contient l'information de la géométrie qui constitue le maillage (exemple: *tubeR.geom*).

```
Geometrie lGeo;  
lMsg = lGeo.importe(lNom);
```

Chacune des entités qui constitue le maillage, par exemple l'entrée, la sortie et les bords d'un tube, sont contenus dans un fichier (exemple: *tubeR.ent*) qu'il faut lire. On assigne ensuite chacune des entités au maillage. Ces entités permettront de fixer les conditions limites sur la frontière de la géométrie du maillage.

```
ListEntitesGeometrique lListeEntite;  
ChaineCar lNomFichierEntites = lNom + ".ent";  
ifstream lFichierEntites(lNomFichierEntites.c_str());  
lMsg = lListeEntite.importe(lFichierEntites);  
lListeEntite.asgnMaillage(lMail);
```

Ensuite, on déclare le type d'élément qu'on utilise. Dans le cas de l'élément P2-P1, décrit au chapitre 2, on prendrait un champ vectoriel 3D avec des fonctions d'interpolation quadratique (*ChampVect3DQuad*) pour le champ de vitesse *lU* et un champ scalaire avec des fonctions d'interpolation linéaires (*ChampScalLin*) pour le champ de pression *lP*.

```
ChampVect3DQuad lU(lMail);  
ChampScalLin lP(lMail);
```

Il faut aussi déclarer une classe gérant le traitement des conditions aux limites. On fait la lecture du fichier (exemple *tubeR.CL*) contenant les fonctions qui définissent les conditions sur chacune des entités du maillage.

```

ListeConditionsLimites lListeCondLim(lChampGeo, lListeEntite);
ChaineCar lNomCL = lNom;
lNomCL += ".CL";
ifstream lFichierCL(lNomCL.c_str());
lMsg = lListeCondLim.analyse(lFichierCL);
lListeCondLim.appliqueChamp();

```

On déclare maintenant des objets de type Form permettant de calculer chacun des termes de la formulation variationnelle telle que décrite à la page 17. Les classes FormStokesNumVitesseNewGen, FormDivDiscrete et FormMatriceMasseLin permettront donc de calculer les différents termes qui formeront respectivement les matrices A_i , B et M telles que décrites au chapitre 2 (la matrice M servant à préconditionner les systèmes $M^{-1} \mathcal{A}_i \delta P = M^{-1} G_i$).

```

FormStokesNumVitesseNewGen    lFormVitesse;
FormDivDiscrete                lFormDiv;
FormMatriceMasseLin           lFormMasse;

```

Pour calculer les différentes intégrales de la formulation variationnelle, on utilise les quadratures de *Gauss* sur des triangles ou des tétraèdres. On passe en paramètre soit le nombre de points de *Gauss* (ex: DouzePtsInternes), soit le degré de précision du schéma d'intégration (ex: 2).

```

SchemaIntg lSchemaIntg;
PtIntgTriangleStd lPtIntgTriangle(PtIntgTriangleStd::
                                   DouzePtsInternes);
PtIntgTetraStd    lPtIntgTetra(2);
lSchemaIntg.asgnIntegration(lPtIntgTriangle);
lSchemaIntg.asgnIntegration(lPtIntgTetra);

```



```

lVectResiduPression.asgnDimension(lMatriceDiv.
                                reqIndicesDepartLigne());
lVectCorrectionVitesse.asgnDimension(lMatriceVitesse.
                                    reqIndicesDepartColonne());
lVectCorrectionPression.asgnDimension(lMatriceMasse.
                                    reqIndicesDepartLigne());

```

PETSc (*The Portable, Extensible Toolkit for Scientific Computation*) est une librairie d'algèbre linéaire numérique qui permet entre autres de résoudre des systèmes d'équations linéaires. Puisque tout est déjà programmé, il suffit d'affecter les fonctionnalités de *PETSc* à notre tâche. Pour ce faire, on conserve des références de type *PETSc* sur nos différents objets de type *MatriceGlobale* et *VecteurGlobal*.

```

MatricePETSc& lA = lMatriceVitesse.reqMatrice();
MatricePETSc& lB = lMatriceDiv.reqMatrice();
MatricePETSc& lM = lMatriceMasse.reqMatrice();
VecteurPETSc& lResiduVitesse = lVectResiduVitesse.reqVecteur();
VecteurPETSc& lResiduPression = lVectResiduPression.reqVecteur();
VecteurPETSc& lCorrectionVitesse =
                                lVectCorrectionVitesse.reqVecteur();
VecteurPETSc& lCorrectionPression =
                                lVectCorrectionPression.reqVecteur();

```

On utilise les solveurs de *PETSc*. Notamment le solveur gradient conjugué avec préconditionneur de *Jacobi* pour résoudre les systèmes $J^{-1} A_i W = J^{-1} R_{1,i}$ (voir l'algorithme d'Uzawa, à la page 31),

```

SysLinPETScGradientConjugué lSolveurVitesse(lA);
PrecondPETScJacobi lPrecondA(lA);
lSolveurVitesse.asgnPrecond(lPrecondA);
lSolveurVitesse.asgnTolerances(lTolRelativeV, lTolAbsolueV,
lTolDivergenceV, lMaxIterationsV);

```

et un solveur direct (par décomposition LU) pour résoudre le problème intermédiaire $A_0 w = v$ présent lors du calcul de la matrice \mathcal{A}_i en faisant du *matrix-free*.

```
SysLinPETScDirect lSolveurPression(lM);
```

On assemble la matrice masse M

```
lMatriceMasse.debutAssemblage();
lFormMasse.assembleMatrice(lMatriceMasse);
lMatriceMasse.finAssemblage();
```

Puis on déclare la matrice de classe `MatricePETScUzawa`, classe spécialement conçue pour gérer la résolution (à l'aide du *matrix-free*) des systèmes concernant la matrice $\mathcal{A}_i = BA^{-1}B^t$ selon les trois étapes décrites à la page 29.

```
MatricePETScUzawa lMatriceUzawa(lP.reqNbTotalDDLs(),
                                lP.reqNbTotalDDLs());
lMatriceUzawa.asgnSolveur(lSolveurVitesse);
lMatriceUzawa.asgnMatricesVecteurs(lMatriceVitesse, lMatriceDiv,
lVectResiduVitesse, lVectCorrectionVitesse,
lListeCondLim);
```

On préconditionne le système $M^{-1}\delta P = M^{-1}G_i$

```
PrecondPETScMatriceMasse lPrecondMasse(lMatriceUzawa);
lPrecondMasse.asgnSolveur(lSolveurPression);
SysLinPETScGradientConjugué lSolveurGC(lMatriceUzawa);
lSolveurGC.asgnPrecond(lPrecondMasse);
lSolveurGC.asgnTolerances(lTolRelativeP, lTolAbsolueP,
lTolDivergenceP, lMaxIterationsP );
```

On entre maintenant dans le cœur de l'algorithme d'Uzawa, décrit à la page 31.

```
Entier lNbIter = 0;
while(lNbIter < lMaxIterations) {
```

On assemble la matrice A_i et le vecteur résidu $R_{1,i}$

```
lMatriceVitesse.debutAssemblage();
lVectResiduVitesse.debutAssemblage();
lFormVitesse.assembleSysteme(lMatriceVitesse,
                             lVectResiduVitesse);
lMatriceVitesse.finAssemblage();
lVectResiduVitesse.finAssemblage();
```

On impose les conditions limites dans la matrice A_i et dans le vecteur résidu $R_{1,i}$

```
lListeCondLim.appliqueMatrice(lMatriceVitesse);
lListeCondLim.appliqueResidu(lVectResiduVitesse);
```

On résoud le système $J^{-1} A_i W = J^{-1} R_{1,i}$

```
lSolveurVitesse.resoudre(lResiduVitesse, lCorrectionVitesse);
```

et on met à jour la correction W dans le champ lU

```
lVectCorrectionVitesse.appliqueCorrection(lU);
```

On assemble la matrice B (seulement à la première itération) et le vecteur résidu $R_{2,i}$.

```
if(lNbIter == 0) {
    lMatriceDiv.debutAssemblage();
    lVectResiduPression.debutAssemblage();
    lFormDiv.assembleSysteme(lMatriceDiv, lVectResiduPression);
    lMatriceDiv.finAssemblage();
    lVectResiduPression.finAssemblage();
}
else {
    lVectResiduPression.debutAssemblage();
    lFormDiv.assembleResidu(lVectResiduPression);
    lVectResiduPression.finAssemblage();
}
```

On vérifie si $\|R_{2,i}\| < \epsilon$

```
DReel lNormeResiduPression = lResiduPression.reqNormeInfinie();
```

```
if(lNormeResiduPression < lTolerance)
break;
```

On peut maintenant résoudre le système $M^{-1} \mathcal{A}_i \delta P = M^{-1} G_i \equiv -M^{-1} R_2(U_i^*, P_i)$,

```
lResiduPression.multScalaire(-1.0);
lSolveurGC.resoudre(lResiduPression, lCorrectionPression);
```

puis mettre la pression à jour ($P_{i+1} = P_i + \delta P$)

```
lVectCorrectionPression.appliqueCorrection(lP);
```

Il ne reste qu'à incrémenter le compteur lNbIter.

```
++lNbIter;
}
```

On écrit les solutions en vitesse et en pression dans des fichiers (exemples: *tubeR.sol.U* et *tubeR.sol.P*)

```
lNomFichierChamp = lNom + ".sol.U";
ofstream lFichierSolutionU(lNomFichierChamp.c_str());
lMsg = lU.exporte(lFichierSolutionU);
lNomFichierChamp = lNom + ".sol.P";
ofstream lFichierSolutionP(lNomFichierChamp.c_str());
lMsg = lP.exporte(lFichierSolutionP);
```

On exporte les résultats dans un fichier (exemple: *tubeR.pie*) dans un format propre au logiciel de visualisation *Vu*.

```
ExportVU lExportVU;  
lExportVU.asgnChampGeometrique      (lChampGeo);  
lExportVU.ajouteChampVect3DContinu (lU,  "U");  
lExportVU.ajouteChampScalContinu    (lP,  "P");  
lExportVU.asgnDegreInterpolation(Exportation::MAILLAGE_LINEAIRE);  
lExportVU.exporteInfo(lNom+ ".StokesNewGen_Uzawa");  
}
```